.EXE

£3.20

The Software Developers' Magazine

DECEMBER 1991 VOL 6 ISSUE 7

Virtual Reality - A new LSD?
We look to the future and the exotic.

So what is VR all about?
Full story with pics.

Behind the tinsel of the next GUI -
A programmer's perspective of Windows 3.1

Yet more Borland C++, now with optimiser.

Sounds peculiar from the humble PC.
Two packages and DIY instructions.

Vol 6 Issue 7 .EXE December 1991

Administration & Subscriptions: Rena Gibbs, Sandy McDonnell, Marianne, Ian & Nutmeg
Reader Services & Promotions: Helena Adams
Advertising Executive: Marc Warren
Assistant Advertising Manager: Ed Butcher
Group Advertising Manager: Sandra Inniss-Palmer
Design & Layout: Mark English
Production Manager: Katerina Adams
Trainee Staff Writer: Cliff Saran
Staff Writer: Paul Kemp
Editor: Willie Watts
Front Cover Illustration: Chris Duggan

Happy Christmas Everybody!

General
.EXE Magazine is independent and not affiliated to any vendor of hardware, software or services. It is published by Process Communications Ltd, 10 Barley Mow Passage, Chiswick, London W4 4PH.
Tel: (Advertising/Editorial/Production) 081 994 6477
(Subscriptions) 0442 824501
Facsimile: 081 994 1533 Telex: 8811418 SPACES G
ISSN: 0268-6872

Subscriptions
.EXE Magazine is a monthly journal for software developers. It is available only by subscription, at a cost of £35 per annum (11 issues) in the UK. 'A subscription implies that this journal will be sent to the subscriber until one of the three expires' - AG Macdonell. The magazine is published around the 26th of the month preceding the cover date. There is no January issue. To subscribe or obtain details of overseas/academic rates, please call 044282 4501, or write to The Subscriptions Manager, .EXE Magazine, 10 Barley Mow Passage, Chiswick, London W4 4PH. We can invoice your company or take ACCESS and VISA cards. In case of a query about your subscription, please call 044282 4501.
Back issues are available at a cost of £3.50 each - call our Chiswick office (081 994 6477) for a list of issues still in print.

Editorial
Editorial enquiries should be addressed to The Editor, .EXE Magazine, 10 Barley Mow Passage, Chiswick, London W4 4PH. We welcome letters, opinions, suggestions and articles from our readers. Please write for a copy of our Contributors' Guide.
Information contained in .EXE is believed to be correct. If errors are found, we will endeavour to publish a clarification in the next available issue.
From time to time, we offer to copy the PC program code described in an article onto diskette for our readers. In this case, please send a blank, formatted disk with a self-addressed, prepaid mailer to the editorial address given above. We can copy both 5.25" and 3.5" disks.
The publishers can accept no liability for any consequences of using software distributed in this way.

Advertising
If you are interested in advertising in this magazine, please write to the Display Advertising Manager/Recruitment Sales Manager (as appropriate), .EXE Magazine, 10 Barley Mow Passage, Chiswick, London W4 4PH, or call 081 994 6477 for details of our advertising rates.

Pronunciation
The name of .EXE Magazine is pronounced to rhyme with 'not sexy magazine'.

Issue theme: Back to the Future

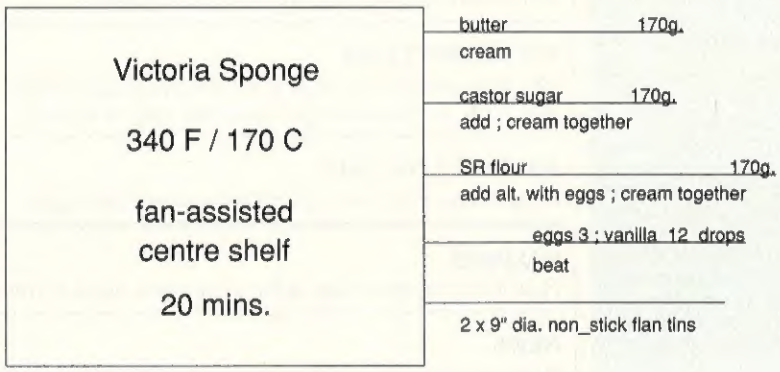
Kitsch 1.0

Emrys Edwards presents a new programming language for recording recipes.

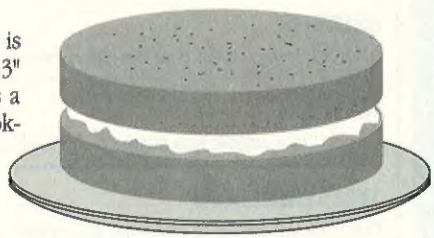
A few years back, I had to learn to cook in a hurry: the cook, my wife, had left. The hardest facts to find out (I thought the most essential) were the cooking conditions: temperature, oven position and time. In the course of my work I had learnt many complex subjects, but still found it difficult to understand a recipe and especially to remember it. However, once I had used one, remembering it was fairly easy. I realised that the difficulty lay in the way the information was presented in the 'standard works' on cookery.

Having read Tony Burzan's excellent book[1] and laughed at his witty TV presentations for the 'Open University', I was already familiar with the idea that the brain stores information in the form of patterns, and not as 'linear text', as in a text-book. Shown in his book is a form of recording ideas called (by me) 'bubble diagrams', or 'neural-maps', if you are trying to impress your colleagues.

Taking in all the above, I decided to sketch out a programming language for recording recipes and called it 'Kitsch'. The syntax is simple and very easily learned. There is no need for a course stretching over two or more weekends, and costing £1200, not forgetting the VAT and hotel accommodation. The only 'operating system' you require, to paraphrase a former Headmaster and Latin scholar, 'is supplied by the intelligence of the reader.'



To write down a recipe, a vertical rectangle is drawn in the centre of a piece of paper, or 5" x 3" (or 8" x 5") index card. The rectangle also has a symbolic significance[2,3] meaning 'on-line' or 'cooking receptacle'.



Syntax

1) Written inside the rectangle is the name of the recipe, and details of the oven environment: fan-assisted, centre shelf, 170° C (340° F).

2) Every 'process' is shown[4] as a line. If the line is attached to the rectangle, the operation is carried out in the receptacle, usually a mixing-bowl or food-mixer. If more than one operation or ingredient is written on a line, the individual operations or ingredients must be separated by semicolons.

3) Any line not attached, signifies an off-line operation: such as beating eggs in a bowl before adding them to the mixture.

4) Each line is read[5] from left to right, the first 'token' being the ingredient (margarine, butter, castor sugar, etc) followed by the quantity.

5) Below the line is recorded the operation to be carried out: cream, knead, blend etc.

6) As in a normal program, 'instructions' are carried out sequentially, starting at the top right hand of the rectangle, and working downwards (or anti-clockwise).

7) There are no library functions to learn, as in C.

To illustrate the method, I have included a 'thumb-nail sketch' of the recipe for the 'Victoria Sponge'. In the Delia Smith 'Bible'[6] this takes up a page and a half.

Method

To prepare a recipe in Kitsch-format, the rectangle is first drawn, and the name of the recipe and the cooking conditions written down inside it. You must first analyse the recipe in order to understand it, and then list, in time order, the various operations to be carried out. These are written down on a series of horizontal lines, starting from the top of the right hand vertical side of the rectangle. Having done this, it is a simple matter to add in the ingredients and quantities above the lines, and the operation(s) to be carried out below the line. The last line usually refers to the cooking receptacles eg '2 x 9" diameter non-stick flan-tins'.

For clarity, the 'Victoria Sponge' recipe has here been created using a Windows 3 DTP system, but normally you would write it down by hand. Looking back on several years of using this method, the remembering comes from first understanding, and then reorganising, the way the recipe information is presented. The 'picturegram' is merely a convenient way of encapsulating this.

The foregoing shows how a little analytical thought, and the application of modern learning theory, can solve an age-old problem. On the other hand, it just goes to show how a computer engineer can make even the simplest subject look complicated.

EXE

Emrys Edwards cooks and programs in Stoke-on-Trent.

References:

(1) BUZAN, Tony. 'Use Your Head' BBC Publications, 1974, 1982, 1984 ISBN 0 16552 9 (paperback) pp. 86-115 (Based on a series of ten television programs by Nancy Thomas.)

(2) YOURDON, E. 'Techniques of Program Structure and Design' (1975) Prentice-Hall.

(3) WIRTH, N. 'Program Development by Stepwise Refinement' Comm ACM 14, 221-7

(4) HOARE, C.A.R. 'Proof of Correctness of Data Representation' (1972) Acta Inforatica 1, 271-81

(5) AHO, A.V, JOHNSON, S.C. 'LR Parsing' (1974) Computing Surveys 6, pp. 99-124

(6) SMITH, Delia 'Delia Smith's Cookery Course' BBC Publications, 1979, 1980, 1981 ISBN 0 563 16365 8 'Classic Victoria Sponge' Part II pp. 265-7

GREY MATTER

Prigg Meadow, Ashburton, Devon TQ13 7DF

Prices do not include VAT or other local taxes, but do include delivery in the UK and Europe. Please check prices at time of order as ads are prepared some weeks before publication. This page lists some of our products– call us for a complete price list and details of discount structure. ORDER BY PHONE WITH YOUR CREDIT CARD.

TEL: (0364) 53499

FAX: (0364) 53071

PROLOG LANGUAGE

Arity Standard Prolog	DOS	£82
Arity/Prolog Comp. v6	DOS	£515
Arity/Prolog Int. v6	DOS	£220
Educ.Prolog (ADA)	DOS	£28
FS Prolog (ADA)	DOS	£45
VMI Prolog (ADA)	DOS	£75
VML Prolog (ADA)	DOS	£135
LPA Prolog Prof Prog Ed	DOS	£300
Flex Pack Prog Edition	DOS	£750
micro-Prolog v3.1	DOS	£69
PDC Prolog	DOS	£240
Prolog++ Pack Prog Ed	DOS	£375
Arity/Prolog Comp	OS2	£745
Micro-Prolog v3.1	CPM-86	£69
Micro-Prolog v3.1	CPM-Z80	£69
LPA Mac Prolog v3	MAC	£450

C LIBRARIES & TOOLS

Comms

C Asynch Manager	DOS	£135
Essential Comms	DOS	£180
Greenleaf CommLib	DOS	£205
Silvercomm C Asynch	DOS	£140

Database

Btrieve	OS2,Win,DOS	£390
C/Database Toolchest	DOS	£45
C-Index/Plus	DOS	£175
C-tree Plus	DOS	£350
Codebase 4.21	DOS	£180
CQL	DOS	£270
D-ISAM	DOS	£325
db-VISTA DBMS Object	DOS	£425
Essential Btree	DOS	£110
Faircom Toolbox Prof v2	DOS	£680
Pro-C & Workbench	DOS	£799
SoftFocus Btree/ISAM	DOS	£80
Sycero.C Network	DOS	£895

Graphics

Essential Graphics Kernel	DOS	£130
Essential Graphics GUI	DOS	£105
GFX Graphics	DOS	£80
GraphiC 6.0	DOS	£305
graphics-MENU & Data Entry	DOS	£180
GSS GDT	OS2,DOS	£425
GX Graphics	DOS	£120
GX Effects	DOS	£120
GX Text	DOS	£90
HALO Professional 2.0	DOS	£270
HGraph	DOS	£80
MetaWINDOW	DOS	£165
MetaWINDOW/Plus	DOS	£220
PCX Programmer's Toolkit	DOS	£150
Real-Time Graphics & M/CT	DOS	£145
TEGL Windows Toolkit II	DOS	£70

Screen

C-Scape + Look&Feel	OS2&DOS	£390
C-Worthy & cwArchitect	DOS	£389
Greenleaf Datawindows	DOS	£215
MEWEL Window System	DOS	£195
MEWEL Toolbox	DOS	£165
Panel Plus II	DOS	£260
ProtoView	Win	£750
TesSeRact CXL	DOS	£42
Vermont Views V3.0	DOS	£325
Vitamin C V4.0	DOS	£250

General & Sundry Libraries

C Tools Plus/6.0	DOS	£105
CodeRunner	DOS	£140
Essential C Utility Lib	DOS	£140
Greenleaf Functions	DOS	£125
Heap Checker	DOS	£30
Heap Expander 3.0	DOS	£85
Hold Everything	DOS	£130
PDK #1, #2, #3	DOS	£70
PowerSTOR	DOS	£85
Resident-C	DOS	£135
TesSeRact TSR Dev Sys	DOS	£42
Turbo C Tools/2.0	DOS	£105
WKS Library	DOS	£160

Math & Scientific

C/Math Toolchest & Grafix	DOS	£45
C Lang Sci Lib	ANY C	£160
Huge Virtual Array & NAT	DOS	£215
Mathpak 87	DOS	£85
Science & Eng Tools	DOS	£79

Tools

C/Utilities Toolchest	DOS	£45
C-Clearly	DOS	£85
Clear+ for C	DOS	£140
CodeCheck Prof	DOS	£435
LALR Pro 4.2	DOS	£275
MKS LEX & YACC	DOS	£165
PC-Lint	OS2&DOS	£85
PCYACC Prof	DOS	£435
Source Print	DOS	£75
Tree Diagrammer	DOS	£75

LOW PRICES FOR MICROSOFT & BORLAND PRODUCTS

WITH FULL TECHNICAL SUPPORT

Microsoft Basic PDS 7.1	**£222**
Microsoft C PDS 6.0	**£222**
Microsoft C & Windows SDK	**£335**
Microsoft Cobol 4.5	**£435**
Microsoft Fortran 5.1	**£205**
Microsoft Macro Assembler 6.0	**£75**
Microsoft Pascal 4.0	**£139**
Microsoft QuickBASIC 4.5	**£50**
Microsoft QuickC 2.51	**£50**
Microsoft QuickC & Assembler	**£83**
Microsoft QuickC for Windows	**£95**
Microsoft QuickPascal 1.0	**£50**
Microsoft Source Profiler	**£45**
Microsoft Visual Basic	**£95**
Microsoft Windows 3.0	**£70**
Microsoft Windows SDK 3.0	**£225**
MS Win Multimedia Dev Kit	**£225**
MS Win MDK & SB Pro Card	**£365**
Borland C++ 3.0	**£206**
Borland C++ & AFX 3.0	**£295**
Borland Corporate Prod Pack	**£330**
Turbo C++	**£80**
Turbo C++ & Turbo Vision	**£69**
Turbo C++ for Windows	**CALL**
Turbo Debugger & Tools	**£89**
Turbo Pascal 6.0	**£69**
Turbo Pascal Professional 6.0	**£137**
Turbo Pascal for Windows	**£103**
Paradox 3.5	**£340**
Paradox Engine 2.0	**£223**

DEBUGGING TOOLS

Bounds Checker	386	£220
Multiscope for DOS	DOS	£140
Multiscope for Windows	Win&DOS	£270
Periscope I/512K	DOS	£490
Periscope II	DOS	£150
Periscope II-X	DOS	£130
Quaid Analyser	DOS	£130
Soft-ICE	386	£340
Soft-ICE & Bounds Checker	386	£470
Turbo Debugger & Tools	DOS	£89
Multiscope for OS/2	OS2	£315

ASSEMBLERS

MS MASM PDS v6	OS2&DOS	£75
PharLap 386IDOS Ext SDK	DOS	£330
PharLap 386IASM/Linkloc	DOS	£830
Optasm	DOS	£125
Turbo Debugger & Tools	DOS	£89
2500AD Z80 ASM	CPM-Z80	£70
SLR Z80ASM	CPM-Z80	£40
SLR Z80ASM-PLUS	CPM-Z80	£140
SLR MAC	CPM-Z80	£40
SLR MAC-PLUS	CPM-Z80	£140
SLR 180 (64180)	CPM-Z80	£40
SLR 180-PLUS (64180)	CPM-Z80	£140

Not all assemblers are supplied with a linker.

PROGRAMMING TOOLS

Ada Compilers	**Algol Compilers**
Assemblers & Libs	**AWK**
Basic Compilers	**Basic Interpreters**
Basic Utilities	**Basic Libraries**
C Compilers	**C Utilities**
C Interpreters	**C Libraries**
C++	**Cobol Compilers**
Comms.Libraries	**Cross Assemblers**
Database Libs.	**Debuggers**
Dis-assemblers	**Editors**
Forth	**Fortran Compilers**
Fortran Libraries	**Graphics Libraries**
Icon	**Linkers/Locaters**
Lisp	**Logo**
Modula-2	**Object Oriented**
Pascal Compilers	**Pascal Libraries**
Prolog	**Rexx**
Screen Libraries	**Smalltalk**

We stock many items for which there is no space in these advertisements.

THE C LANGUAGE

Refer also to the C++ section.

C COMPILERS

NEW Watcom C 8.5/386 supports 32-bit Windows & DOS apps, 32-bit DLL's, includes royalty free DOS/4GW extender with 32Mb Virtual Memory, new debugger, etc all at a SPECIAL INTRODUCTORY PRICE of £495. CALL US NOW.

NEW QuickC for Windows includes QuickCase:W, image & dialog editors, Windows hosted environment, etc. If you want to write Windows programs, BUY ONE NOW. We're waiting for your call.

Aztec C86-p	DOS	£120
Aztec C86-d	DOS	£175
Aztec C86-c	DOS	£275
High C 1.61	Win&DOS	£390
Hi-Tech C	DOS	£165
Microsoft C 6.0	OS2&Win&DOS	£222
MS QuickC 2.5	DOS	£50
MS QuickC & QuickAsm	DOS	£83
MS QuickC for Windows	Win	£95
Mix Power C & lib s'ce	DOS	£30
Topspeed C Std	DOS	£110
Topspeed C Prof	Win&DOS	£165
Watcom C 8.5	OS2&Win&DOS	£295
High C 386 Local 1.7	PL386	£585
High C 386 Global 2.3	PL386	£650
Hi-Tech C 386 PDS	386	£385
Intel 386 C Code Builder	Win&386	£455
Watcom C 8.5/386	Win&386	£495
Topspeed C Prof	OS2	£165
Aztec C II-p	CPM-80	£120
Aztec C II-c	CPM-80	£200
Hi-Tech C	CPM-Z80	£105
Mix C	CPM-Z80	£35
Aztec C68K MPW C	MAC	£110
Aztec C68K/Am-P	AMIGA	£120
Aztec C68K/ST-P	ATARI	£120
Hi-Tech C 68000	ATARI	£140
Prospero C	ATARI	£55

We supply 2500AD, Avocet, Aztec, Lattice, IAR and Hi-Tech Cross Compilers hosted on DOS and targeted on Z80, 6502, 6801, 68HC11, 6301, 6809, 7811, 8051, 8096 & 68000/020. Please call for information or advice.

C INTERPRETERS

New V5 Instant-C is MS C V6 compatible.

C-terp V3.5	DOS	£165
Instant C V5	DOS	CALL

DISK COPYING SERVICE

We can copy files to and from 600 disk formats including CP/M, CP/M-86, MS- DOS, PC-DOS, APPLE, SIRIUS, BBC, TORCH, APRICOT, HP-150, TRSDOS, AMSTRAD, ATARIST, MACINTOSH.

Our charge is £10.00 + disk + VAT with discounts on small quantities and disks are normally despatched within 24hrs of receipt.

For more information call us.

ADA COMPILERS

IntegrAda	DOS	£645
OpenAda DOS	DOS	£260
OpenAda DOS 286	286	£935
OpenAda DOS 386	386	£1865

CROSS ASSEMBLERS

We supply cross-assemblers by Avocet, 2500AD, Crash Barrier (METAi) and IAR Systems hosted on DOS, CP/M-86 and CP/M-80 with over 30 target processors. In total over 200 products with no space to list them here. We hold some stock but you should allow 10-14 days for delivery. Please call for information or advice.

C/C++ Show

A chance for developers to come face-to-face with some of the more important vendors and find out what's really happening to C and C++. The C/C++ Show will be held in West London on Tuesday 28 January, 1992. Admission is free, but numbers are limited, so it is advisable to book a place ASAP. Ring Judi Holly on 0622 691616.

UNIX Bargain

BSD/386 is a new version of UNIX for the 386, based on the Network Software Release 2 of Berkeley UNIX. It provides complete functionality, including TCP/IP networking, X Windows and Sun's NFS file system. There is also an ANSI C and C++ compiler. The complete system (including source code and executables) only sets you back £625. BSD/386 will start shipping in January 1992 and will be distributed in the UK by Berkeley Software Design International(Europe) Ltd on 0227 781675.

HyperDoc

Version 4.1 of SourceDoc from Intelligent Solutions, a hypertext documentation tool for programmers, is now available. SourceDoc 4.1 (previously PolyDoc) includes an input filter (with source code) which automatically inserts key words to automate documentation. SourceDoc runs under DOS and OS/2 and is priced at £259 for a single user or £995 for a 5-user network version. Contact Readmar Systems Ltd tel 071 6255255.

Acting dBASE

PowerLibW, from Boston-based ETN Corporation, now provides an interface to Whitewater's Actor 4.0. PowerLibW is a library of more than 90 functions that allow access to dBASE and Clipper-compatible database files. Support is included for multiple indexing, multiple relations, filters, memo fields, expression evaluation and much more. PowerLibW retails for $295 and can be purchased from ETN Corporation on 0101 7174352202.

Compiler Offer

Pecan Software is offering its UNIX compilers at a special promotional price of £295 each. The product range includes UCSD Pascal, Modula-2, Fortran 77 and C compilers. There are also a number of assemblers available. The Pecan UNIX tools are offered for several systems including SCO UNIX, Interactive UNIX and Sun3. For more information contact Pecan Software Europe Ltd on 0272 586650.

Long live the x86!

A new pact between IBM and Intel should ensure that Intel's 80x86 chip architecture is (for better or worse) still with us in the 21st century. The 10-year technology agreement means that the companies will work together on a series of projects to design powerful microprocessors based on the x86. The main thrust of development will be to sweep more and more computer functions (such as memory controls, graphics arrays, cache memory and systems bus controls - including MCA) into the chip itself. It is expected that the first products will be available to the industry within about two years.

To undertake this work, IBM and Intel have established the Noyce Development centre in Boca Ralton, Florida (named after the late Dr Robert N. Noyce, semiconductor pioneer and co-founder of Intel). The centre will be staffed by engineers and scientists drawn from existing IBM and Intel laboratories around the world.

In addition, IBM has secured the right to manufacture a portion of its requirements for standard 486s, as well as future products in the Intel x86 line. In recent years IBM has exercised rights, under a previous technology licensing agreement with Intel, to manufacture part of its i386 requirements and to develop independently new products derived from the 386 architecture. The 386SLC, designed with an internal 8 KB cache and announced in October, is such a derivative product, and is reported to offer a significant performance boost above standard 386SX chips.

Windows NT SDK

Microsoft is now releasing the first version of the Windows 32-bit Development Kit, incorporating an alpha version of its Windows NT operating system. Designed to run on 386 and 486-based PCs, the the software will be available to ISVs and corporate customers in the UK. Representing nearly two million lines of code, the kit offers a complete 32-bit development platform with Windows 3.1-compatible user interface and integrated LAN Manager services. Microsoft is not charging for this release of the development kit, or for related support.

UK-based ISVs interested in receiving the SDK should fax details of their development programmes to Microsoft on 0734 507624. Mark your application papers for the attention of the Windows NT beta programme Systems Marketing division. Microsoft plans to widen distribution next year with subsequent releases of the kit, building up to final shipment of Windows NT in 1992.

Embed With MS C

Designers of embedded system applications can now use Microtec's 80x86 family of XRAY debuggers and emulators with the Microsoft C compiler to develop their code, thanks to CrossLink, a new product from Microtec research.

CrossLink enables developers to link their own start-up code (eg for setting up the Stack Pointer and Segment registers) with applications compiled using Microsoft C. There are also run-time libraries and I/O routines to convert MS-DOS code to embedded applications.

The Xray86 debugger can then be used to debug the resulting code. Microtec has also released an in-circuit debugger, Xray/Monitor, which enables the real time debugging of embedded code on the target system. Xray86/Monitor can be either downloaded to the target system or blown into EPROM.

CrossLink costs £240 and is distributed by Microtec Research on 0256 57551.

NetWare-Oracle connectivity

West Midlands-based LAN specialist, Firefox Communications, has announced support for Oracle's SQL*Net Version 2. Distributed Oracle database systems and Oracle SQL front-end programs (such as Oracle Card for Windows 3) will now be able to operate over Novell NetWare systems and use OSI protocols. The development is based on Firefox's NOVOS system for NetWare, which provides OSI communications facilities as NetWare Value Added Processses (VAPs) or NetWare Loadable Modules (NLMs), and on Oracle's SQL*Net product set. It is designed to allow Oracle products using SQL*Net to communicate across NetWare LANs using Novell's IPX/SPX protocol then, via a NetWare server running Firefox's NOVOS relay system, to OSI-based mainframes and UNIX host systems that support Oracle RDBMS products. This means that Oracle-based information systems can now be distributed across NetWare LANs and inter-networks, without the need for each workstation to support an OSI protocol stack.

Firefox NOVOS systems supporting SQL*Net V.1 are available now, with support for V.2 in Q1 '92. Prices for the NOVOS SQL*Net gateway systems will start at £695. Contact Firefox on 0675 467244.

SURPRISE PACKAGE

With Borland's new Turbo Pascal for Windows® you can develop professional-looking Windows applications quickly, and easily, even if you've never programmed for Windows before.

Surprisingly Fast

Turbo Pascal for Windows includes Borland's new Object Windows® application framework free! So, you can now develop Windows applications fast because pre-defined objects for windows, menus, dialogs, controls and more are already provided.

Surprisingly Easy

Turbo Pascal makes programming Windows applications easy because it incorporates a Windows Integrated Development Environment. And with the Whitewater Resource Toolkit,® to give you a head start, you've got everything you need to easily create Windows programs.

Surprisingly Good Value

Because Turbo Pascal for Windows has everything you need to develop powerful Windows applications it's unnecessary to buy the Microsoft Windows Software Development Kit (SDK).®

In fact, for only **£149.95** (plus VAT) Turbo Pascal for Windows gives you surprisingly more and costs a lot less than other Windows development systems.

To existing Borland language users we're offering a number of great upgrade deals. For more information call **0800 378880** now or contact your regular Borland dealer. He'll be anything but surprised to hear from you.

From the makers of Borland C++,® Turbo C++,® Turbo Pascal,® Paradox,® Quattro Pro,® ObjectVision® and Sidekick.®

Please send me the FREE Turbo Pascal for Windows Information Package.

NAME: _____

COMPANY: _____

ADDRESS: _____

_____

POSTCODE: _____ TEL: _____

I currently program in Pascal ☐ Basic ☐

I am a Borland Language user ☐

Please return to: Borland Turbo Pascal for Windows, Borland International (UK) Ltd, FREEPOST RG1571, Twyford, Berkshire RG10 8BR. .EXE 12/91

BORLAND
SOFTWARE
KNOW HOW

Borland International (UK) Ltd.
8 Pavilions, Ruscombe Business Park, Twyford,
Berkshire RG10 9NN.

CIRCLE NO. 334

Multi-lingual CodeBase

Version 4.5 of Sequiter Software's xBASE library supports access to dBASE-compatible data from C, C++, Turbo Pascal for Windows and Visual Basic. The library enables programmers to work directly with the data, index and memo files of dBASE, FoxPro and Clipper from DOS or Windows. The latest FoxPro 2.0 and dBASE IV index file formats are now also supported. CodeBase 4.5, with complete multi-user source code, retails for £225. Existing users of CodeBase or CodeBase++ can upgrade for £95. Contact The Software Construction Company on 0763 244114.

GUI for Clipper

VI is the Visual Interface library for Clipper, from Bits Per Second (tel 0273 727119). It allows Clipper developers to build a Windows-style GUI for their DOS applications. The library provides high-resolution windowing with interactive controls, an integrated event-handling mechanism, and standard window and control objects. It is designed to work with Clipper 5.01. VI will be available from January 1992 and will cost £245.

dB Access

System C has launched a new file handler for its Sycero C program generator. dB Access is designed to replace the CodeBase 4 library from Sequiter Software, which currently allows Sycero-generated C applications to access .DBF files. In benchmark tests conducted by System C, dB Access has shown a performance improvement over CodeBase 4 ranging from a factor of three to a factor of 10. It is claimed that the product uses an improved algorithm for retrieving data and reading indexes and is also designed specifically for use with Sycero C. dB Access costs £195 and is available direct from System C on 0622 691616.

Dublin-based Glock/USL C++

Glockenspiel has been nominated as the sole European Marketing Agent for Unix System Laboratories' (USL) C++ V3.0 source code (see .EXE Nov 91 'news'). 'Our strategy has always been to provide added value to USL's C++', said John Carolan, Managing Director of Glockenspiel, '...the new source code agreement allows us to provide our customers with an initial customisation of C++, while leaving them free to carry out their own upgrade work.' USL may be contacted on 081 5677711. Glockenspiel is on 010353 1 733166.

Latest MultiView

JSB MultiView Desktop V3.1 is the latest version of SCO's integration tool for Microsoft Windows. It enables users to run multiple DOS, Windows, Xenix and UNIX sessions within separate windows. Along with full file transfer protocols, data may be copied and pasted between DOS and UNIX windows. Links may be set up between a DOS and a UNIX application using the DDE. In addition to LAN Manager for UNIX, this version provides several mediums for connection with remote Xenix, SCO UNIX or SCO Open Desktop hosts, includes RS232, NetBios and TCP/IP. UNIX network printers may also be accessed from DOS or the Windows Print Manager. Using X11/AT and TCP/IP, X Windows servers can be accessed from DOS. MultiView provides configuration facilities which enable an X application to be invoked by simply specifying the name of the executable - MultiView loads the X Server and starts the X client. In fact, MultiView allows the user to select which terminal emulation mode should be used with which application or login (eg running an X application next to VT100 emulation).

With only a single login, the user can open up to six UNIX sessions. Along with the many new features that have been added in V3.1, MultiView now includes a fully configurable, 'point and click' icon driven desktop which enables UNIX applications to be launched using familiar Windows controls. A single user license for JSB MultiView Desktop V3.1 costs £225. For more information contact The Santa Cruz Operation on 0923 816344.

Smelly fax?

Zetafax from Equisys is a software package which lets PC network users send faxes directly from Microsoft Windows applications. According to the manufacturer, the product enables users to send by fax exactly what they see on the screen. 'This technology is starting to be called WYSIWYF - What You See Is What You Fax' said Equisys MD, Chris Oswald. The software is designed to be independent of fax hardware, and currently supports Swedish Telecom and Panasonic fax machines. Touchbase's Worldport 2496 fax modem should be next to join the list. Drivers for various fax cards are planned for early 1992. Zetafax runs on a range of PC networks, including NetWare and LAN Manager.

Prices start at £895 for a 5-user entry-level system (excluding fax hardware). There is also a C-language API available for £495 for developers who wish to interface with the fax-sending software. For more information, contact Equisys on 071 4032227.

GForce upgrade

Calypso software has released version 2.2 of its GForce graphics library for Clipper. The new version features a GUI-driven screen editor (forms designer), allowing the developer to create, edit and test screen designs in the GForce graphical environment. The screen editor generates the appropriate Clipper code for inclusion in the final application, or alternatively the screen file may be directly executed using a library function, providing a graphical equivalent to dBASE .FMT files. There is also improved mouse support. GForce V2.2 costs £185 from QBS Software on 081 9944842.

New X Windows & Motif

Metro Link Inc has released Metro-X V1.5, a complete implementation of the MIT X Windows System release 11.4 which boasts increased performance and support for UNIX System V Release 4 (SVR4). It is able to take full advantage of SVR4's Dynamic Shared Libraries - this means that the size of the client code is reduced significantly. The Metro-X Development System includes OSF/Motif libraries, IMAKE, XView libraries and the OpenLook windows manager. The GNU C compiler is also provided.

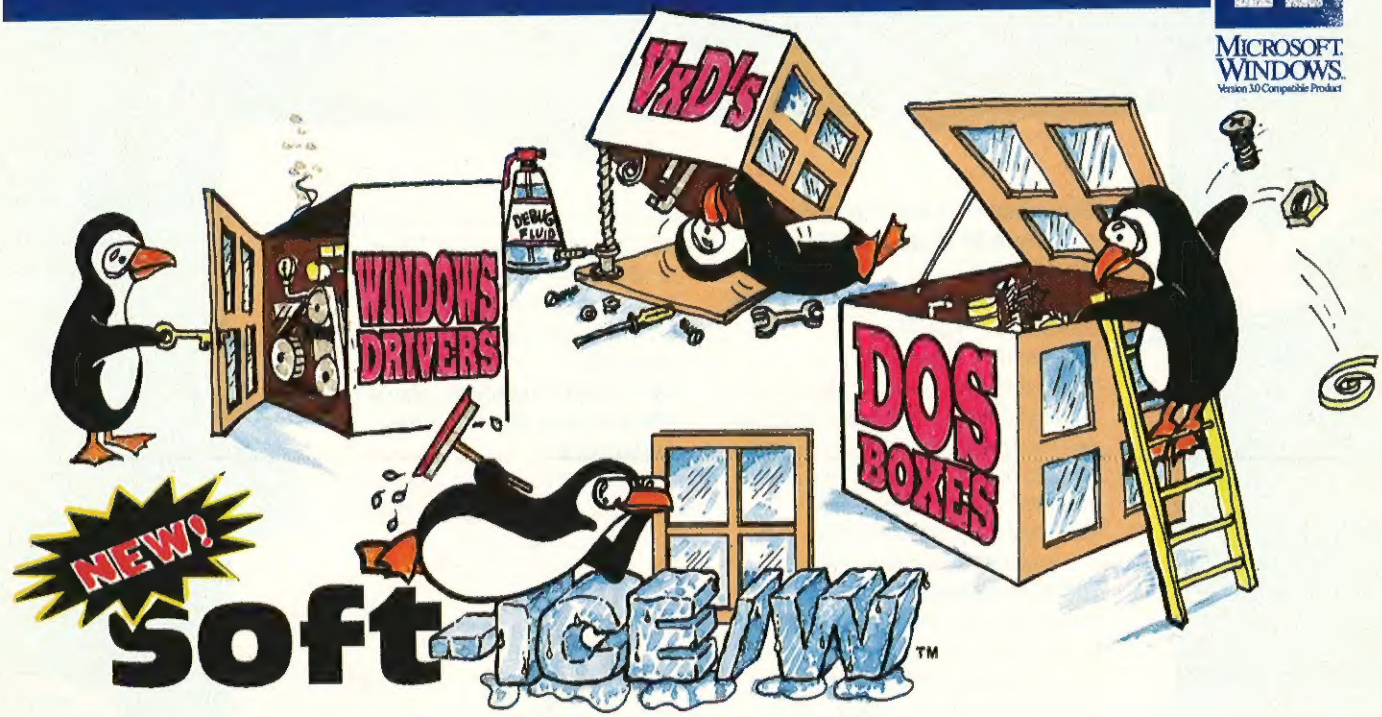
Sun Motif V1.1.3 is Metro Link's latest implementation of OSF/Motif for the Sun. The development package contains several libraries including the Motif library, Motif Resource Manager, User Interface Library and the Widget Creation Library. Metro-X V1.5 costs $299 and Sun Motif V1.1.3 costs $199. Call Metro Link Inc on 0101 (305) 9707353 for more information.

Maths Classes?

Matrix.h++ and Linpack.h++ are two new C++ maths class libraries from Rogue Wave Software Inc. They are both totally compatible with Tools.h++ and Math.h++ (also from Rogue Software). Linpack.h++ provides all the functionality of its Fortran cousin. Matrix.h++ is a subset of Linpack.h++ and provides methods for tackling matricies including vectors, statistics, complex numbers and Fast Fourier Transformations. The libraries are available for most computer systems. The MS-DOS version of Matrix.h++ costs $199 ($399 with source). Linpack.h++ is priced at $299 for the MS-DOS version ($495 with source). Rogue Software can be contacted in the US on 0101 503 7572311.

Get Inside WINDOWS!

MICROSOFT WINDOWS
Version 3.0 Compatible Product



Debug Windows at the systems level!

Soft-ICE/W takes you inside Windows! Debug and explore with power and flexibility not found in any other Windows debugger! Soft-ICE/W allows you to debug at the systems or applications level or simply learn the inner workings of Windows.

- Debug VxD's, drivers and interrupt routines at source level
- Debug interactions between DOS T&SR's and Windows Apps
- Debug programs in DOS boxes
- Display valuable system information
  (from the total memory occupied by a Windows application, to the complex internal structures of Windows)

Soft-ICE/W uses the 386/486 architecture to provide break point capabilities that normally require external hardware. Nu-Mega, which pioneered this technology with the introduction of its award winning Soft/ICE for DOS, now gives Windows programmers the same debugging power... *and still at a software price.*

Own the debugger that combines the best "view" of Windows internals with the most powerful break points of any software debugger.

Soft-ICE/W . . . Only $386

ALSO

CodeView for Windows users

See what you're debugging without flash

CV/1 version 2.0 runs CodeView in a graphics window letting you see CodeView and the program screen at the same time. If you're spending more than 10 minutes a day in your debugger you should be using CV/1.

- Runs CodeView on **any** monitor that supports Windows.
- No more "Flip-Swap-Flash". Step through code without the annoying flash.
- Small Window Mode shrinks the debugging window to a few lines of source.

***CV/1* v2.0 . . . $129**

Call: **(603) 889-2386**
FAX: **(603) 889-1135**

P.O. Box 7780
Nashua, NH 03060-7780 U.S.A.

Nu-Mega™ TECHNOLOGIES INC

RISK = NULL
30 DAY
MONEY-BACK GUARANTEE

CIRCLE NO. 335

Early booking discounts available

Your chance to meet the most experienced Clipper minds in the world

Awed by Arrays? Troubled by TBrowse? Confused by Code Blocks? Overawed by Objects?

Spend three days with Nantucket and the best Clipper minds in the business and these subjects will become as clear as day. This year the Annual European Clipper Developers Conference will be held at the Birmingham Metropole Hotel on **20/21/22 January 1992.**

Choose the sessions of most interest to you from over 26 on offer, whether you're a newcomer to Clipper or an old hand! Many of the best known speakers and authors on Clipper will be there to help you with tips and tricks, including: Matt Whelan, Rick Spence, Jud Cole, Greg Lief, Philip de Lisle and members of the Nantucket Development Team.

There will also be special previews of some powerful new technology for Clipper.

An exhibition of companies supplying add-ons and services for Clipper will be held during the conference and delegates will be able to use a Computer Lab to practise their new found skills.

As a Clipper developer, can you afford to miss this inspiring conference?

Act Now! Call the Conference Administration for your brochure and more information:
Telephone: 071-706-3295
Fax: 071-706-3296.

THE THIRD ANNUAL EUROPEAN CLIPPER DEVELOPERS CONFERENCE

CIRCLE NO. 336

THE THIRD ANNUAL EUROPEAN CLIPPER DEVELOPERS CONFERENCE

20/21/22 January 1992 Birmingham Metropole, National Exhibition Centre, Birmingham

Terminal PM

A new tool from Intelligent Environments is designed to ease the conversion of existing 3270/5250 terminal-based applications to run under OS/2 Presentation Manager. AM/HostView automates the programming process for screen capture and update of terminal applications from within a PM program. HostView allows online point-and-shoot selection of fields and rows of data, giving them logical names and then generating code for the company's Applications Manager (AM) design tool. The product works by examining the existing terminal application and then calculating the screen co-ordinates, attributes and exact function calls required to read or update the data. HostView then creates reusable modules that perform the host screen interaction.

AM/HostView costs £3,000 per workgroup and operates in conjunction with AM (the Developers' Edition is priced at £5,000). Intelligent Environments is on 081 9406333.

A ROOM with a GlobalView

Rank Xerox's ROOMS, previously seen only as a prototype, is now available in product form. ROOMS extends the GUI concept of the electronic desktop to encompass a whole electronic office building. Or, as the company puts it, 'ROOMS is the co-ordination of multiple virtual desktops with linking doors allowing the user to work on many projects concurrently with each project preserved just as you left it. In real life this is equivalent to providing every employee with several offices, each designed for a particular job'. Far out.

ROOMS is now provided as a standard feature of the new Rank Xerox GlobalView UNIX version which runs on the company's 6540 SPARC-based workstation. For further information, contact Roger Bearman at Rank Xerox on 0895 251133.

New multi-tasking OS

AMX is a real time multi-tasking operating system from Kadak Products Ltd, Vancouver. Designed for developers of embedded 80x86 and 68000 applications, the AMX kernel provides preemptive priority scheduling with optional time-slicing. Separate managers control message passing, semaphore signalling, resource allocation, event synchronisation and memory management. AMX is compatible with popular development tools from Microsoft, Borland, MetaWare, Phar Lap, Watcom, Avocet, Intermetrics and Paradigm. The Insight Debug Tool co-operates with source-level debuggers such as Turbo Debugger, CodeView and Paradigm DEBUG/RT to simplify system testing. Until now the product has not been available in the UK. There are no royalty fees to pay and the product comes with source code. The 80x86 version sells for £1,950 and the Insight debugging tool costs £995. Contact Great Western Instruments on 0761 452116.

British Dance

US-based software house GUIdance Technologies is now distributing its Choreographer GUI development tool in the UK. The product provides a graphical development environment for creating Windows 3 and OS/2 PM applications. It is designed mainly for front-ending client/server applications and building pretty interfaces to boring old character-based mainframe applications. The Choreographer interface can be driven by applications written in C or COBOL and can call code written in those languages. LU6.2 and APPC protocols are fully supported along with a variety of SQL databases. Choreographer is priced at £10,000 per workstation and is available from Software Generation on 0462 422525.

Powersoft PowerBuilder

PowerBuilder v1.0 from Powersoft is a new graphical development environment for GUI-hosted SQL-based client/server applications. Version 1.0 runs under Windows 3 and supports several SQL servers, including Microsoft/Sybase SQL Server, Gupta's SQLBase and Oracle. Future releases are planned which support other windowing environments and popular relational database servers (eg HP's ALLBASE/SQL). The graphical point-and-click environment is augmented by a (naturally) object-based 4GL, PowerScript.

There is support for all standard Windows objects and a SQL 'Smart DataWindow', which is a custom window object for database manipulation (without using SQL) that integrates the application with the back-end database. There is also support for DDE and the ability to call C functions in a Windows DLL from the scripting language.

Prices start at £1,900 for a standalone version and £2,900 for a version that supports a networked database. There is also a run-time licence fee of £100-£190 per distributed copy.

PowerBuilder is available in the UK from Admiral Software Ltd on 0276 692269.

Papers please!

The UK Computer Measurement Group (UKCMG) has issued a final call for papers for its 7th annual conference on information technology. The conference themes will be Multi Vendor Platforms and Value for Money in IT. The event is due to take place on 5-8 May, 1992 at the Brighton Metropole Hotel. If you have any ideas, contact Keith Allen on 0444 247423. Alternatively call the UKCMG office on 0753 522204.

From PC to X

You can now turn your PC into an X-terminal by plugging in Inmos's new iX card. This transputer-based graphics card uses a 25MHz T400 transputer with 1 MB of video RAM to produce an X-server which is compatible with MIT X11R4. It supports the standard VGA/SVGA screen modes, although higher resolutions can be attained using a high resolution monitor. iX costs £850 and is available from Inmos on 0454 617910.

dBASE Browser

Do you ever need to glance at a database when you are in the middle of producing a report or developing an application? Do toast crumbs in the bed make you itch? db.Quick is an invaluable memory resident utility which enables you to do just that. Using pop-up screens, it allows the user to view records or perform database searches while inside another application (such as a word processor or spreadsheet). db.Quick costs £69.00 and is distributed by Alpha Software Corporation on 0752 606881.

Multimedia PC

MiroMovie is a new Windows compatible multimedia PC, complete with software. The hardware consists of an 800 x 600 multi-media audio/video card together with a 20" colour monitor. The software includes screen drivers and image processing tools which run under Windows. An optional graphics controller is also available which provides support for 1024 x 1280 screen resolution in 16/24 bit colour. For more information, please contact Ambitron on 0635 36555.

A Crimbo Message

To all our readers and advertisers, we extend our thanks for your support throughout the year, and wish you all a spiffing Christmas and wizard New Year. Don't forget that .EXE has a month off in January. Meanwhile, enjoy the seasonal revelry and see you next February.

Letters

We welcome short letters on any subject that is relevant to software development. Please write to The Editor, .EXE Magazine, 10 Barley Mow Passage, Chiswick, London W4 4PH. Unless your letter is marked 'Not for Publication', it will be considered for inclusion on this page.

Crocodile Tears

Sir,

Your peculiar attitude towards UNIX has finally driven me to write. Wake up to the nineties; the era of everybody and his dog (sorry, DOS), migrating to UNIX. What is .EXE's attitude at this time? UNIX, it would seem, is too difficult for .EXE to handle, so you carp on about how obscure it is. So, why the patronising attitude to UNIX? I can only assume that it's an irrational fear of the dreaded UNIX command line.

What prompted me to write was your news article *(.EXE ,September '91)* 'Environmentally Friendly'. Gosh! Xtree even works with any terminal that supports `TERMCAP` (your capitals, why?), providing, of course, that it has a screen display of 80 columns by 24 rows. As any half-decent programmer will tell you, any program which doesn't use `termcap` is not coded properly (if it has to use fancy screen stuff, that is).

Golly! File backup has been greatly simplified. No more hours trying to work out how to back-up my system. What a treat.

Soon it will be easy as under DOS (Ha Ha!). I could go on (and on, and on...) but you probably get my drift. Stop being so patronising and demeaning towards UNIX, and start taking it seriously. As a programmer who is more than casually acquainted with DOS, I can't think of an environment that I'd rather use than UNIX. Wake up .EXE, the UNIX revolution is here!

Graham Nicolls
Primar Ltd
Camberley, Surrey

'TERMCAP' was copied, with mind in neutral gear, from the Xtree company's press handout - Ed

Taking the Hype

Sir

I was flattered and gratified that Richard Pickard found my item sufficiently stimulating to base a 'Soapbox' article on it. I am dismayed, though, that in spite of all messages he discerned, one of the more important ones appears to have passed him by.

Dr Pickard discussed indexing, attribution, scanning, and the like, as if hypermedia can aspire to be no more than an appendix to what we do already. There is no point in this; books perform these functions perfectly well, and all one can get by following this route is books that must be plugged in. He is careful to distinguish between teaching and learning, but in so doing appears to neglect entertainment, persuasion or exploration.

Descending through a tree of linkages (Dr Pickard's implicit model) is appropriate only for the organisation of facts; other kinds of information require different structures. My article was an argument (partly logical and partly rhetorical) and the inverted tree I used appeared to be an appropriate form.

I am astonished that he regards the mental and physical stimulation of TV to be unbalanced; to my eyes very little TV provides either! In contrast, practically all the feedback I have had demonstrated curiosity about the uncompleted links. I think this represents a degree of success for the experiment and an encouraging vindication of the technique. Dr Pickard is quite correct to say that this stuff demands practice, but it seems to me that there is no point developing these skills unless we're going to get something back. By all means be critical, but please don't let that get in the way of imagination and vision. And don't worry - if it doesn't work, it will die.

Jules May
Herts

More on QEMM

Sir

I would like to comment on the 'Quarterdeck effect'. There is no doubt that QEMM and DESQView are excellent products; the best of their kind, in my experience. The problem is that, as they say in Pisa, you can't build a skyscraper on marshland.

`OPTIMISE` notwithstanding, one can have hours of good clean fun fiddling with one's TSRs, searching for that 'holy grail' of enough memory to do anything useful, while retaining a measurable half-life between reboots. Here are a couple examples of some of the holes I have fallen down over the last few months:

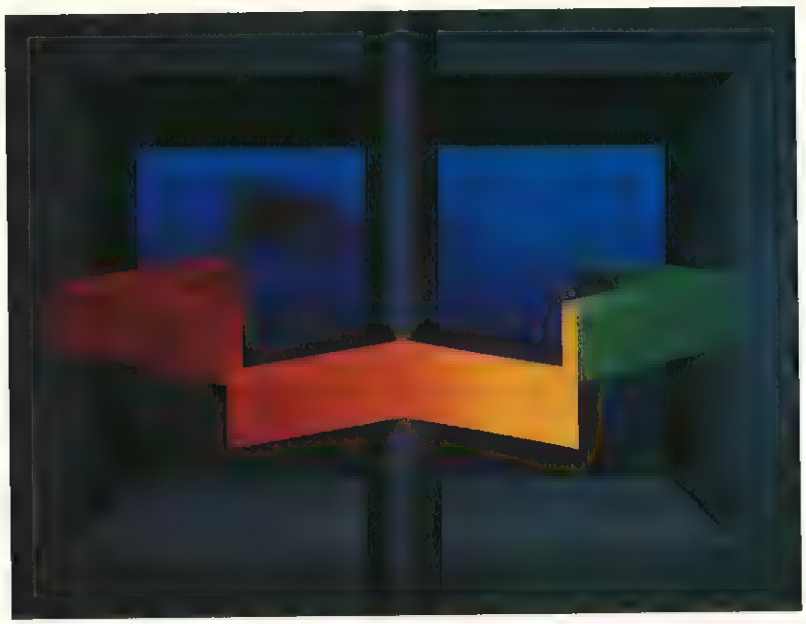
- QEMM V6.0 'stealthy memory' causes `cc:Mail` to crash (but only on my machine; my colleagues evidently possess a better class of rabbit's foot).
- The same setup screws up CodeView in Extended memory mode.
- Something keeps crapping on my environment (still, I guess one could say that about the world in general).
- When our UNIX machine unilaterally decides to log me out, my PC dies (it wasn't always this way - maybe my di-lithium crystals have cracked).
- If I load my IPX driver high and I'm running short of high memory, the UMB chain gets corrupted (unless Libra is in conjunction with Virgo).

What this comes down to is that, as a software developer, I am increasingly aware that DOS's limitations are becoming more than just irksome. DOS is surely living on borrowed-time. If OS/2 V2.0 lives up to all the IBM promises, we should all be in a position to get down to some real work.

Paul Sanders
SilverPlatter Information Ltd
London

Letters submitted to this page may be edited. The writer of the best letter of the month, as judged by the Editor, will be rewarded by a T-shirt or similar-valued .EXE trinket. The best letter is the one printed first.

Who put C++ to Work?



glockenspiel
class constructors

Glockenspiel C++

Glockenspiel C++ leads the field in object-oriented programming. It's the most effective and efficient implementation around. It conforms exactly to the AT&T C++ specification – and they don't come any closer than that! Portable too: it's on more platforms than any other C++ implementation.

Glockenspiel CommonView®

Glockenspiel CommonView is the leading C++ class library for developing Windows, PM and OSF/Motif applications. Logical, well structured and easy to use, CommonView is 3 to 5 times more productive than the API.

C++ Training & Consultancy

Glockenspiel C++ training courses reflect our experience as C++ developers. From design to development, *learn from the best!*
Our C++ consultancy services match your needs as you commit more resources to C++ and object-oriented development.

Further Information

GLOCKENSPIEL, 39 Lower Dominick Street, Dublin 1, Ireland. +353 (1) 733166. Fax +353 (1) 733034

- **US: IMAGESOFT**, (516) 767-2233. Fax (516) 767-9067 **OASYS**, (617) 862-2002. Fax (617) 863-2633
- **UK: QA TRAINING LTD.**, (0285) 655888. Fax (0285) 650537.
- **Italy: INFERENTIA**, (02) 26680568. Fax (02) 2364258.
- **Sweden: LINSOFT**, (01) 3124780. Fax (01) 3152429.
- **Germany, Switzerland, Austria: PSI**, (06021) 492-0. Fax (06021) 492-112
- **Benelux: RIJNHAAVE**, +31 (71) 218121. Fax +31 (71) 216118

Virtual Worlds

Virtual reality is in the process of moving from Sci-Fi to actuality. Al Roth pulled on his eyephones and his datagloves, and went and found out how far it had got.

If you have managed to avoid reading anything about Virtual Reality (VR) then I'll wager you live on a lesser Hebridian island. Most folk I know are sick of hearing about it. But despite all the hype, I believe there really is something big going down. Signs are good that within the lifetime of the average .EXE reader we will all be designers of, or players within, virtual worlds that we have created. (Cue vision of gangley, acneyed youth running along sun-kissed tropical shores hand-in-dataglove with software constructs of Darryl Hannah and Kim Basinger).

Unfortunately we are still a long way from this dream; therapeutic as it may be. Indeed for home use we are still a long way from *any* kind of VR at all. However, as we shall see, systems are now emerging which may bring this prospect much closer.

What and who?

First, what is VR, and who is doing it? According to Bob Stone, VR is 'the generation, using computer graphics, of realistic three-dimensional visual, audio, and tactile worlds in which a suitably-equipped user can explore and interact with virtual objects using natural human skills'. He should know, as he is the Deputy General Manager at the Advanced Robotics Research Centre. Managed by Advanced Robotics Research Limited (ARRL), the centre was set up in June 1988 with £5 million start-up funding from the Department of Trade and Industry. The Human Factors Research Programme began at the ARRL in 1989, and is concerned with the design of human-system interfaces for the supervision and tele-operation of advanced robotics systems.

The Total Immersion Approach to Virtual Reality

Virtual reality is seen as a cornerstone of ARRL's research work. In particular, the group has just developed a new tactile feedback glove - 'TELETACT' - designed to work with existing interactive glove controllers such as VPL Research Inc's DataGlove. Stone expects to be launching a product based on the combined ARRL/VPL technologies in Japan this month.

But not all ARRL's effort is so down to earth. Currently one of biggest VR unknowns is the effect of prolonged exposure to cyberspace on human physiology. The ARRL is planning a series of psychological tests which will attempt to determine the recovery period following exposure to VR. It has been suggested, for instance, that individuals should not drive immediately following use of VR, since human sensory apparatus adapts to things; spatial effects need to wear off. Put another way: maybe you can get used to being able to move through walls without hurting yourself! Another exotic application: a team of psychiatrists is to visit ARRL to discuss the possibility of using VR to undertake phobia research. VR allows researchers to control a patient's exposure to the appropriate phobia-inducing situations, say open spaces, spiders or snakes. The author has shared a virtual reality room with an animated spider the size of a Mini, and can vouchsafe for the disconcerting effect.

The Next Generation of Workers may toil in Virtual Offices

Budget VR

Companies are beginning to emerge with the aim of bringing VR to a larger user base by making the technology more available on inexpensive stock hardware. One such product is WorldToolKit, a library of C routines aimed at developers of VR applications. Developed by US company Senses8 Corporation, and available in the UK from Virtual Presence Ltd, the WorldToolKit system imports 3D models from popular file formats such as DXF, STL and ASCII and renders them in real-time.

The company says that WorldToolKit allows the user to manipulate objects within models, and animate choreographed sequences. Lights, texturing, graphics and images can be applied in real-time, which Virtual Presence claims reduces design time, and allows instant viewing of different virtual worlds.

WorldToolKit interfaces to a range of I/O peripherals including the Spaceball 2003 interactive input device, VPL's Eyephone LX colour stereo head-mounted display screen, the Flight Helmet from Virtual Research and the Position Tracker from Logitech. Virtual Presence says that the next release will be the inclusion of the 'Upfront' 3D perspective sketching and drawing environment. WorldToolKit runs on 386/486 under DOS and is priced from £4000.

Bristol company Division is looking at the development of VR applications based on its transputer-based Provision system. Provision is based on a distributed memory architecture in which a number of processing 'clusters' are assigned responsibility for a given task. Each cluster has its own local memory and a control processor (Inmos transputer) for controlling any specialised peripheral hardware such as D-to-A converters. The system also incorporates other processors. The 3D geometry (clipping, lighting and so on) is handled by Intel i860, taking advantage of that chip's floating point maths capabilities. The z-buffering and the actual writing of pixels (a purpose for which the i860 is apparently ill-suited) is performed by a Toshiba HSP card.

Provision can convert popular file formats, such as DXF files, into an internal format, and then allow the user to view objects and program them directly in C. Provision has a number of library calls for creating, deleting, moving and manipulating objects around the system. The facilities consists of basic elements such as lighting, and a set of actors which provide a 'service' - such as displaying the image, or creating an appropriate sound.

The software is object-oriented to the extent that discrete computational entities (actors) are each responsible for a given activity, and which can co-operate to achieve a desired result. Division's system makes it possible to assign functions to objects so that they can become autonomous. Each object gets polled at a certain time interval and will then do an iteration of its function. This could be a fairly simple operation - say updating its position - or rather more complex, involving communication with other objects in the system. In addition to information about its displayed position, the object might also have sound attributes, and need to send sound messages out to the sound card. In true OOP fashion, these behaviours can be inherited by descendants. Figure 1 shows a sample of Division's C code responsible for creating a number of simple environmental objects (EnvObjects). Future versions of the system will be coded in the ubiquitous C++.

Techno drawbacks

At the moment, even for the corporate giants, VR is an expensive technology to get into. Most of the technological push to date has been aimed at delivering 'immersion' systems, in which the virtual world is entered by wearing a head-mounted display system, and strapping on the necessary input devices (typically gloves, grips etc). This kind of system typically requires huge computing resources to handle the real-time display. The result is that when you move your head there is a noticeable delay before the virtual image 'catches up'. This

NEW VERSION 3

Extra features include:

Plotter Support ■ WBS & OBS Coding ■ High-Resolution Charts ■ Proposal Format ■ Extended Printing Features ■ End Date Scheduling ■ Extended Importing & Exporting ■ Optional LAN Support

Save £460 when you upgrade to InstaPlan EMS

Now get the leading project management software for just £85!

YOURS FREE
Worth £14.95, this guide is yours whether you keep InstaPlan or not

Whatever your project planning or management needs, you can't make a better choice than the InstaPlan award-winning family of planning software.

Instaplan is widely recognised as offering the best combination of top performance with low price.

Now for just £85, you can upgrade from any existing planning software to InstaPlan EMS – the top-of-the-range InstaPlan package, supporting up to 8MB of expanded memory for full-scale project planning.

The deal is simple. You send us photocopies of your system disks, showing the serial number, and we'll send you InstaPlan EMS for just £85 – a fantastic saving of £460 on the normal selling price.

Supremely easy to use, InstaPlan EMS handles in excess of 10,000 activities and offers multi-project support, integrated work accounting, trend analysis and many other advanced features.

Why not get real planning power at your fingertips? With our 30-day unconditional money-back guarantee you've nothing to lose! For full details of this unbeatable money-saving offer, simply clip and return the coupon below, or call us on 081-908 2423.

InstaPlan

Deepak Sareen Associates,
28 Preston Road, Wembley HA9 8BR.
Tel: 081-908 2423. Fax: 081-904 6668.

Read what the experts say:

"Highly professional, highly competent, tremendous range of planning choice, terrific value-for-money."
GOOD SOFTWARE GUIDE

"InstaPlan is proof that simplicity of design and operation can be achieved without abandoning a good set of project management features."
PC MAGAZINE

". . . Useful . . . intuitive . . . practical . . . well worth the price."
PROJECT MANAGER TODAY

"InstaPlan offers the features of a £2,000 package with none of the trade-offs of its low-cost competitors."
PERSONAL COMPUTER MAGAZINE

"Technically it is extremely competent, being one of the only programs that is capable of levelling resources across multiple projects."
COMPUTER SHOPPER

"So simple to use that you wonder why other packages make a meal of resource allocation."
PC MAGAZINE

"Excellent rethink of requirements for project management packages . . . Ease of use – excellent, functionality and documentation – good."
PC USER

BEST PC MAGAZINE BUY

PERSONAL COMPUTER EDITOR'S CHOICE

WHICH COMPUTER? BEST BUY

DON'T MISS OUT ON THIS SPECIAL UPGRADE OFFER

Complete and return the coupon today for full details on InstaPlan, plus this unbeatable upgrade offer.

Please send me a free information pack on InstaPlan, and details of the special InstaPlan EMS upgrade offer. Plus how to receive my **FREE** guide to 'Getting the job done' (offer closes 31st January 1992).

Name _____

Job Title _____

Company Name _____

Address _____

_____

Postcode _____ Tel _____

EXE 12/91

Do you currently use any project management software?

☐ NO ☐ YES If YES, please state which

_____

POST TODAY – NO STAMP NEEDED

DEEPAK SAREEN ASSOCIATES
FREEPOST, 28 Preston Road, Wembley HA9 8BR

Tel: 081-908 2423 Fax: 081-904 6668

CIRCLE NO. 338

lag is caused by both the head-attitude tracker and the system-induced lag. Rendering the image is a very computationally expensive process, especially for high quality images. It is still probably the rate-limiting step in the entire process. It will be many years before current computer power is able to render photo-realistic images in real-time.

In addition to the delay, the display itself is often grainy, blurred and of rather poor resolution. This is why much of the VR work has focused on better display technology, better input devices, and on throwing more computer resources at the problem to achieve a faster rendering capability. The research into display technology is perhaps the most bizarre. My favourite project is one involving the use of lasers to project a virtual image directly onto the user's retina. Perhaps a case where it might be wise to wait for the technology to mature before buying. I, for one, don't want such a gadget in this year's Christmas stash, even if it is 'theoretically possible to calibrate units without field trials'.

But on top of all these other drawbacks, the artificial world itself is typically very Spartan. Having spent all the money on the gear, you still find yourself in a Virtual Reality desert. Until recently, little work has been done on developing toolkits which support the creation and manipulation of interesting virtual worlds. This is changing.

The Virtual Desert blooms

One company which is taking a fresh approach is UK-based Dimension. Founded in 1983, the company was initially targeted towards entertainment sector, and indeed was involved with the CyberZone TV game which was screened a few months ago. Dimension has devised the Virtual Reality Toolkit (VRT), which is an interactive utility used for the creation of virtual worlds. Designed for use with proprietary Desktop VR system worlds, the product can also run on any other desktop or immersion virtual reality system. The Dimension package also includes a Shape Editor (a real time 3D shape modeller), the World Editor (a real-time 3D environment creation program), and the standard shape and object libraries.

The VRT is under beta-test at a number of sites throughout the UK, including ARRL, which was able to run the system on the Division transputer box. By default, Dimension's kit runs on a PC (although admittedly of a somewhat higher spec than your standard WordPerfect workhorse) 486/33 based unit, 256 KB Cache, SPEA FGA1 'intelligent' graphics card (featuring the Texas 34020 graphics processor), and a 19" high resolution monitor. The effect is not truly virtual, in the sense that you are not immersed in the world. However, as a compromise it's not bad. VRT has a clever way of enabling people to build virtual worlds and allowing humans to 'fly' through them using a 'spaceball' - a three-axis, 6 degree of freedom, joystick.

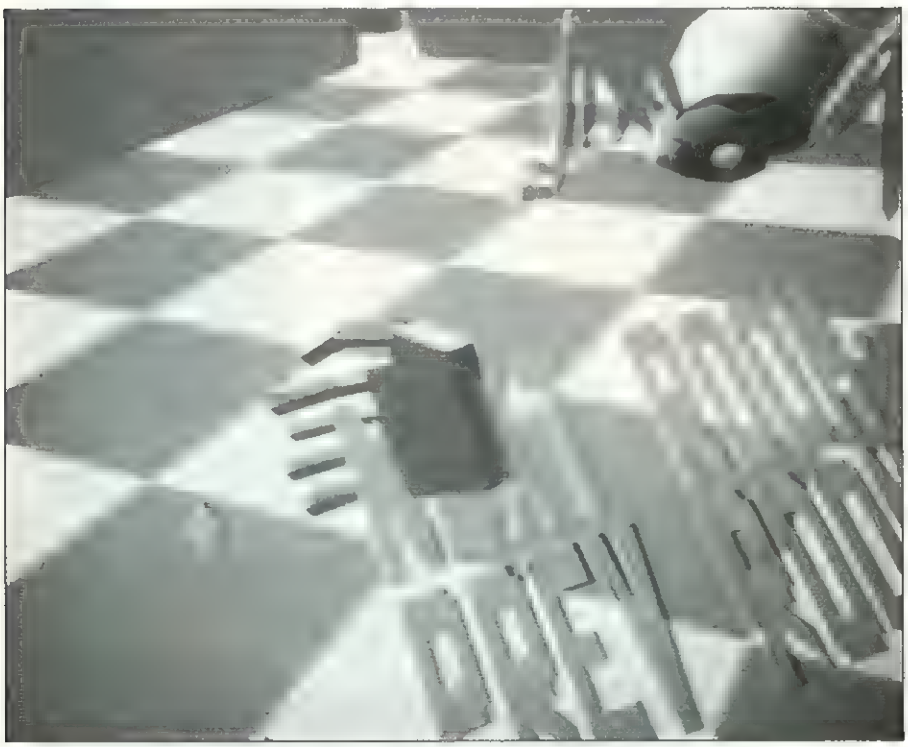
OOP and AI

Objects just get everywhere don't they? Most folks agree that objects are a good idea, but for programming virtual worlds they are exactly the ticket. Object-oriented programming allows for a very neat mapping between virtual entities and underlying software representation. To see why this works so well, you might care to look up a previous .EXE article (June '91) in which I covered computer animation in the context of television adverts, such as the Smarties commercial; the OOP principles offered there apply in exactly the same way to VR. Briefly: once you have described the behaviour of a given construct (say a virtual creature such as a bee) then that same behaviour maps well onto any other instances of bees that you create. Conventionally, if the animator had to generate a sequence of three bees flying along then you would create a function which ensured that they did not collide. If you then decided to double the number of bees - or create a full swarm - a hardwired function would have

to be rewritten to cope. Using OOP technology, each object is responsible for its own collision detection.

In addition to OOP, there are other software techniques which may yet contribute a lot to the VR party. One of the more interesting is AI. Assigning intelligence to virtual objects could well be a fun-filled adventure for AI programmers over the coming decades. Much of the link between VR and Artificial Intelligence is conceptual and more than a little speculative. It is likely that AI will enhance the reality of virtual worlds. Indeed it may be the case that much of the software development environments which will be necessary in order to design and construct virtual worlds will embody AI techniques. This may be particularly true of AI research into model-based reasoning, intelligent agents, planning, and machine learning.

Virtual Reality can be a Virtual Desert



3D Menus offer an escape from the Cyber Spider

Applications

The potential for VR applications seems huge. One sexy use of VR is the provision of artificial laboratories in which scientists are able to explore problems in a manner that closely resembles investigation in a real physical laboratory, but which requires no direct programming. Other applications include the construction of virtual wind tunnels, simulations, CAD, and realistic architectural designs to produce virtual buildings that can be physically explored by the client in cyberspace. The potential for research into hazardous environments seems immeasurable.

One trend that is guaranteed to emerge will be the virtual office. Large companies in the future will save themselves the fortune they currently spend providing, equipping, and heating physical offices. Virtual offices will appear as more companies begin to appreciate the benefits of VR, and sponsor a virtual workplace where workers can have meetings, deliver presentations, and enjoy telepresent R&R together at the end of the working day. Teleconferencing is going to be big business.

VR is at the cusp of a number of converging technologies. These include AI, computer graphics, OOP and highly parallel, real-time computing. Each of these (and more) will need to be integrated to generate truly-interactive, and convincing artificial realities. Nevertheless, it now seems likely that the research of the future may be carried out not in physical laboratories full of test tubes and white coats, but instead by telepresent acolytes interacting within a photo-realistic laboratory.

Goodbye GUI?

So what? Good question. It is my belief that VR will fundamentally change the nature of our world. Indeed it already is doing so. Someone remarked that a good point to make to non-believers is that all our money is already kept in Cyberspace. (Since I don't have any money, this doesn't etc etc.)

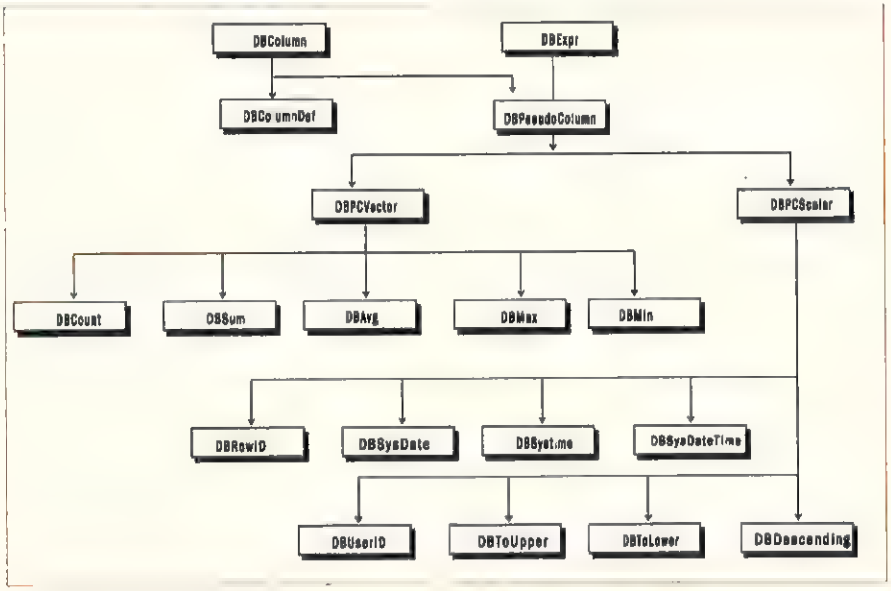
For me one of the most significant (and rather delightful implications) of VR is that gesture-based GUIs are going to die out, perhaps even before they die in. Why is anyone going to want to fool around with windows and mice when you can wear eye-pieces, datagloves, datasuits, and 'immerse' yourself in an alternate reality. Let's face it, it's going to make Windows seem a bit tame.

But there is another thread that I would also like to develop here, based on my belief

CommonBase

Database portability for C++ developers

Write portable database applications the easy way!
Use C++ and CommonBase and you can forget about the differences between APIs. Get the job done while they debate standards for "Dynamic SQL" and an SQL API interface. Harness the power of C++ and let CommonBase manage your application's data conversions and storage details across multiple platforms.
C++and CommonBase – the portable solution to information management.



A sample from the CommonBase class hierarchies

First we brought you Glockenspiel C++ – the most portable C++ compiler around.

Then we brought you Glockenspiel CommonView, the portable class library for developing Windows, PM and OSF/Motif applications.

Now we bring you the solution for portable databases – the CommonBase C++ class library.

Use C++ and CommonBase and forget about the hassles of working with different databases, operating systems and compilers. CommonBase is portable across all three.

It takes care of the differences between the various APIs and the problem of dynamically managing many cursors in Embedded SQL.

The CommonBase class interface for ISAM databases is a subset of the interface to SQL databases. So it's easy to write your application using an ISAM database and later port it to an SQL database – without facing a major rewrite.

Reusable, portable code!

It's what C++ is all about.

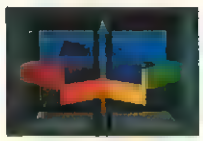
Specifications:

CommonBase requires Glockenspiel C++ or Borland C++.

CommonBase supports Microsoft / Sybase SQLServer, Oracle, Coromandel Integra, Coromandel ObjecTrieve, Gupta SQLBase.

CommonBase contains approximately 35 classes which are organized into separate hierarchies – there is no base "Object" class.

CommonBase documentation inclues tutorial and class library reference manual and a Quick Reference guide to classes and functions.



glockenspiel
class constructors

PUTTING C++ TO WORK

To order:

Glockenspiel, 39 Lower Dominick Street, Dublin 1, Ireland. +353 (1) 733166. Fax +353 (1) 733034.
North America: Imagesoft Inc., 2 Haven Avenue, Port Washington, NY 11050. (516) 767 7839. Fax (516) 767 9067.
UK: *QA Training Ltd.* (0285) 655888. Fax (0285) 650537. ***Italy:*** *Inferentia,* (02) 26680568. Fax (02) 2364258.
Benelux: *Rijnhaave,* +31 (71) 218121, Fax +31 (71) 216118. ***Sweden:*** *Linsoft* (01) 3124780. Fax (01) 3152429.
France: *Microformatic,* (01) 48701900. Fax (01) 48702729. ***Germany, Switzerland, Austria:*** *PSI,* (06021) 492-0. Fax (06021) 492-112.

that most modern GUIs are still really rubbish. Of course there are some very clever hombres doing incredible things with gesture-based interfaces (you know the kind of thing - spiral-shaped pull-down menus, laser-driven or optical mice powered by thought-alone etc). The point is we don't actually want smarter mice/windows/etc. We don't even want gesture-based interfaces. Instead, we need chatty clever computers, like Captain Kirk's. As computers get smarter, with more sensory apparatus built-in, maybe we can expect GUIs - which are, after all, just a huge design compromise - to disappear entirely. I, for one, hope so.

```
#include <VL.h>

int example1 (void)
{
  Environment       E;
  Actor             a;
  EnvObject_Instance teapot1, teapot2, chair;
  EnvObject         object;
  Light_Instance    I;
  Light             ambient light;
  Point             lp;

  /* initialise the Actor */

  a = VLInitActor ();

  /* connect to the "root" Environment */

  E = VLConnect.Environment (a, "root");  /* connect to the root env */

  /* create a single red ambient light */

  ambient_light.light_type = VL_AMBIENT_LIGHT;
  ambient_light.red = 0.5;
  l = VLCreate.Light.Instance (E, &ambient_light);

  object = EnvObject_Initialiser;  /* set up default state */

  /* create two teapots */

  /* position at 100,100,50 */
  lp.x = 100; lp.y = 100; lp.z = 50;
  object.pos.p = lp;

  /* visuals */
  object.view.pos.p = lp;  /* position in Env. space */
  strcpy(object.view.c.name, "..\\models\\teapot.viz");
                                    /* name of .viz model file */

  /* Instance name */
  strcpy(object.c.name,"teapot1");

  teapot1=VLCreate.EnvObject.Instance(E, &object);

  lp.x = 200; lp.y = 250; lp.z = 80;
  object.pos.p = lp;
  object.view.pos.p = lp;
  strcpy(object.c.name,"teapot2");

  teapot2=VLCreate.EnvObject.Instance(E, &object);


  /* create a chair */

  /* position at 300,300,300 */
  lp.x = 300; lp.y = 300; lp.z = 300;
  object.pos.p = lp;

  /* visuals */
  object.view.pos.p = lp;  /* position in Env. space */
  strcpy(object.view.c.name,"..\\models\\chair.viz");
                                    /* name of .viz model file */

  /* Instance name */
  strcpy(object.c.name, "chair");

  chair=VLCreate.EnvObject.Instance(E, &Object);
}
```

Figure 1 - C code for creation of simple environmental objects

VR is bound to be a second nail in the coffin-lid of GUIs, or at the very least is set to change the whole nature of the beast. Within some VR systems it is possible to conjure up a menu from inside the virtual world. What you see is a huge menu that can take up half a room in size. Typical menu options might be 'Fly', 'Grasp' etc. Each option might change the role of the glove - from a means of sailing through virtual worlds, to a hand that can manipulate objects. This technology is with us already, and is improving quickly. Perhaps future GUIs will only be found within virtual worlds.

Yuletide turning

VR is an expensive technology. The good news is that we are now seeing the emergence of companies providing less-expensive, non-immersion VR systems. The bad news is that they are still too dear for Santa to bring us one. At least this year.

But there's no doubt; the world is changing. Maybe one Christmas Eve Santa will arrive on virtual reindeer, and drop our presies off in cyberspace. Imagine the rush on Christmas morning as millions of kiddies tear into virtual reality to unwrap their goodies. *(Creep in Quatermass music.)* No more mountains of discarded wrapping paper, or heaps of broken toys immediately destroyed by exuberant youngsters. Nothing going on in the real world at all - just millions and millions of silent, helmeted individuals wriggling in their cybersuits. *(Music reaches ominous crescendo).* A society where people have literally lost touch with themselves...

Anyone for virtual turkey?

EXE

Al Roth is a journalist who specialises in the flashier side of computer technology. He is the deviser of the Al Roth Diet ('Jog 5 miles a day and eat plenty of fresh chocolate cake') and lives in an amazingly accurate Virtual Reality simulation of Blackpool.

Contact numbers: National Advanced Robotics Research Centre is on 061 745 7384, Division is on 0454 324527, and Dimension can be reached at 0734 810077.

See Jules May's column, elsewhere in this issue, for an alternative view of VR.

BRIEF'S NEW MOUSE SUPPORT GIVES YOU GREATER AGILITY AND CONTROL.

U.K. VERSION
DUAL MEDIA
SUPPORTS DOS & OS/2

More than 100,000 programmers around the world swear by the virtues of BRIEF – its unmatched power, flexibility, and intuitive command structure. Now, thanks to mouse support and other new features, version 3.1 of the best-selling text editor will make you quicker and more productive than ever. ¶ Hit a button on the mouse – a special, user-configurable menu window pops up and lets you execute any command you wish. Mark text and move around in your windows without using the keyboard. Zoom your windows to fill the screen. ¶ As snappy as it is, the mouse is only one enhancement among many. BRIEF 3.1 also takes advantage of EMS memory whenever it reads in a file. The result: faster performance, whether you're editing a file or running a macro. ¶ So what else is new? Redo – a step for step complement to undo. Support for Microsoft C 6.0 Advisor and Microsoft's Programmer's Workbench. And more. ¶ BRIEF's seamless interface with other Solution Systems products gives you instant access to the tools you need. A single keystroke lets you keep track of product development with Sourcerer's Apprentice – Version Control for the Professional. Analyze product performance with CHARGE. And nothing beats dBRIEF for tailoring BRIEF to Dbase or Paradox. ¶ *BRIEF sells for £199 excluding VAT.* ¶ *Once you start working with BRIEF 3.1 you'll really fly.*

READY, AIM, PROGRAM.

Order your copy of BRIEF today.

call:

0763 244141

Solutionsystems
THE PHYSICS OF PROGRAMMING
1 The Maltings, Green Drift,
Royston, Hertfordshire SG8 5DB
Telephone: 0763-244141
Facsimile: 0763-244025

Deskterm™

Discover how you can add a Motif Interface onto Character-based Applications.

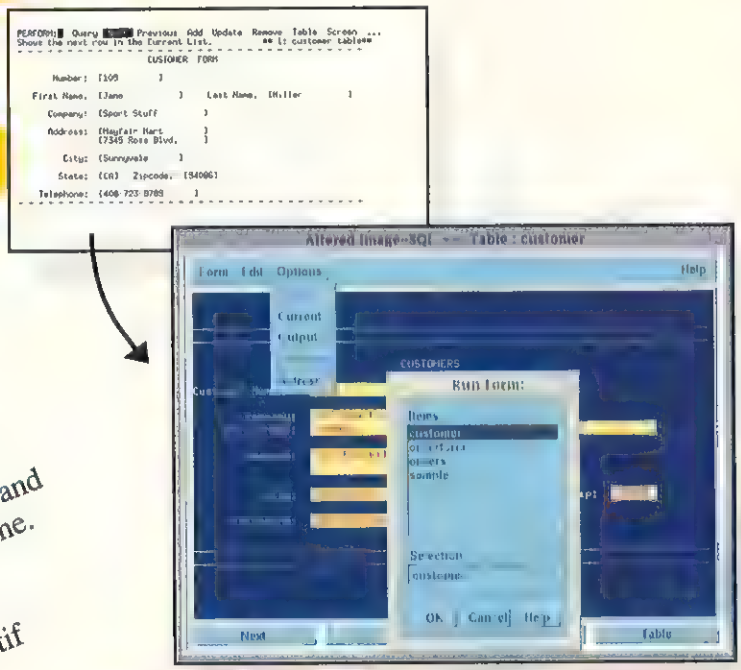
YOUR RECIPE FOR SUCCESS

COOKING TIME

90% saving on re-writing your software for X

METHOD

1. Take existing character-based application
2. Add IXI's Deskterm Software.
3. Run on UNIX system and *simmer* for a short time.
4. Before serving, decorate with Motif



Like many software developers, you may be evaluating how to produce a Motif version of your application. It's no easy decision. For many businesses, the task of rewriting a 20 year old COBOL application or one written in FORTRAN or a 4GL is too costly in terms of time and resources.

IXI has the answer. Deskterm will revolutionize your path to Motif. Experience shows that Deskterm can reduce the time to add a Motif front-end by more than 90%, bringing your application to market years ahead of the competition.

Using Deskterm you can produce a fully featured graphical user interface with pull down menus, scroll bars, dialog boxes, pushbuttons, mouse control and multiple fonts and colors. What's more, you can do all this without access to the application's source code.

You don't need to know anything about X programming (at either Xlib or toolkit level) to move your software to Motif. Deskterm uses existing character-based programming skills allowing you to concentrate your time and money on building applications rather than wrestling with X and toolkits.

UNIXWORLD 1·9·9·0 best PRODUCTS

So if you're looking for a fast and painless route to Motif, contact us today for information on +44 223 462131.

IXI

IXI Limited · 62-74 Burleigh Street · Cambridge · CB1 1QJ · England · Tel: +44 223 462131 · Fax: +44 223 46132

CIRCLE NO. 342

Windows 3.1

Dr Watson was the dull, unobservant half of a certain brace of fictional detectives. So what part could he play in the latest release of a certain GUI? Tony Dodd reveals all.

Windows 3.1 has been made available in pre-release form far more widely than any previous release. Developers are encouraged to ship DLL libraries for Windows 3.1 in advance of the launch of the system itself, so that customers can use the enhanced facilities although they are still running Windows 3.0.

In this article I want to give an idea of what Windows 3.1 will bring to the Windows programmer. Although the beta-test releases are (reasonably enough) protected by non-disclosure agreements, a fair amount of information about the system has been made public either by Microsoft itself at technical conferences or by developers shipping the new libraries. Out of this I shall try to assemble a picture of what the system contains. Charles Petzold's article in *MSJ* was especially useful.

Windows 3.1, as Petzold observes, has been presented as Windows 3.0 plus True Type. In fact, as far as the programmer is concerned, True Type involves very little extra work: there are a whole lot of new fonts available, but the selection mechanism is much the same. But there is a great deal more to Windows 3.1 than that.

¡OLE!

Probably the most significant new feature is OLE (object linking and embedding), which you must pronounce as though it were a Spanish exclamation. While much of Windows 3.1 is still shrouded in the mystery of beta-testing, OLE applications are already being made available, and the characteristic OLE menu items are creeping onto menu bars everywhere.

The purpose of OLE is to allow the user to construct compound documents, that is, documents that contain data of many types. There was a time when, in order to integrate graphics, spreadsheet and word processor facilities you had to buy some monster package from a single manufacturer. If you liked Brand X spreadsheet and Brand Y word processor you were simply out of luck. In Windows, however, there is a data exchange system called the clipboard, and provided both applications ran under Windows you could, with luck, copy a chart from the spreadsheet and paste it into the word processor.

Programmers who have used the clipboard know how that works. The copying application takes the clipboard and places the data on it. Unfortunately one application's data is another one's gibberish, and what actually happens is that the copying application places data in a number of formats; with each format it must either supply a handle to the data or promise to render the data when someone wants it.

A spreadsheet might place its data on the clipboard as a graphical image in bitmap format. If the word processor can incorporate bitmap graphics it will take that data and paste it in. If it cannot cope with bitmaps it might look for text data, but text rendering of a spreadsheet would be a poorer representation. The copying application puts data items on the clipboard in descending order of quality. If you have Excel 3.00, copy a few cells to the clipboard and then look at the clipboard format menu; there is plenty of choice for a pasting application. If you haven't got Excel, look at Figure 1.

At the top of the menu are the weird and wonderful formats that spreadsheets use to exchange data. Suppose the word processor application understands rich text format. Rich text format includes a way of specifying tables, so by taking the clipboard data as a table in rich text format, the spreadsheet structure is maintained. Further down are the disastrously inaccurate text formats; these lose most of the data. But those three items in between, called Native Owner Link, and Object Link, are part of the OLE mechanism.

The native link format is just the spreadsheet range as it would have been stored in Excel's own memory. This, you may think, is unlikely to appeal to the average word processor. However, the Owner Link data is, in effect, a small note saying that Excel takes responsibility for this document, and is willing to look after it via OLE. Excel is the server, the word processor the client. If the word processor couldn't cope with any of the good formats, it can take the native data and keep a record of the server details.

The data is now embedded in whatever document the word processor is working on. This is an important point; there is no reason to suppose that the data will go on existing anywhere else. The spreadsheet from which it was pasted may be deleted. The word processor must take responsibility for the data.

What is the word processor to do with this new and incomprehensible slab of data?

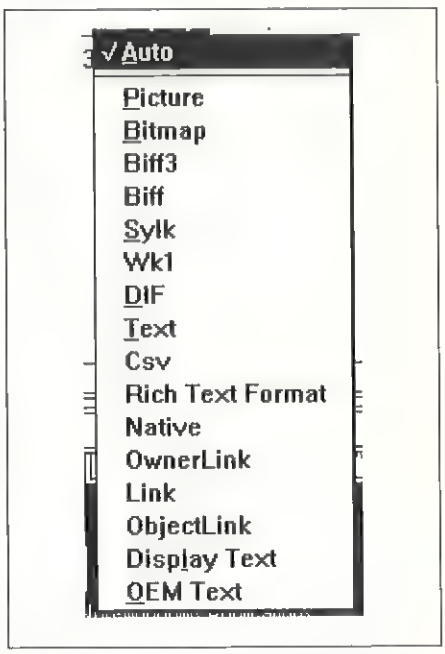


Figure 1 - Format menu of clipboard with Excel 3.0 data

Probably it would like to display it and print it. If the user wants to change it, that must be allowed. And, of course, the data will need to be saved with the parts of the document that the word processor does understand. In general, when the word processor needs something doing to the data, it calls up Excel and has the action performed. Typically it may want the data displayed in a particular HDC, or edited, or saved to a file.

This simple model is worth keeping in mind, though it is not the whole truth. What actually happens is that the word processor looks up Excel in the system registration database and discovers, *inter alia*, that there are a number of verbs associated with the object, that is, things that Excel will do with the object. Typically, but not invariably, these will play and edit the object. Playing an object may just mean showing it; but it could also mean playing a piece of digitised sound or executing a program. The attraction of OLE is that you can have a message to play without knowing at all how it will be played.

Furthermore, the word processor doesn't call Excel directly, but instead calls the client library, which calls the server library, having woken Excel up if necessary, and the server library calls Excel. Somewhere within these libraries a certain amount of special case logic is bestowed. For example, the client library, when it took the native data from the clipboard, will have noticed that there was a bitmap of the data and retained that too, so that requests to display the object can be fulfilled by the client library without the need to awake the server. In OLE applications, you will find a menu item or items that invoke the various verbs associated with embedded objects. Click a different object and the item changes to the appropriate list of verbs.

Object linking

We said that the client takes responsibility for the data embedded in it; but OLE is about linking as well as embedding. A link is an active connection to data stored elsewhere; in the case of the spreadsheet, editing the spreadsheet will change the appearance of the word processor document. Typically the word processor will have an option called Paste Link, meaning that the data should not be embedded but that a link should be created. To create a link, the Object Link data is examined; as before this explains who is responsible for the data.

At this point it is easy to lose track of things. You are almost certainly thinking 'what is the difference between owner and object link?'. I feel obliged to tell you, but you may regret asking. The owner of a piece of data is the program that can perform the verbs advertised on it. Excel spreadsheet data is always owned by Excel wherever it turns up. The source of data is the place where it lives, which may, as we have seen, be in a word processor document. When you put data on the clipboard, you must use the owner link to specify the owner, which may or may not be you. If you copy an embedded object, you just pass on the owner link you received and the native data. If you copy a link, you pass the owner link but not the native data (you don't have it!). But the object link refers to your document, and may be used by to create a link to your document. You are the only person who understands the structure of your documents. If you deposit an object link saying that you are Prolog and the object is the thing in file `foo` that you call `earwig`, then you must be ready to be woken up in the middle of the night by some word processor and asked for earwig, even if `earwig` is an Excel spreadsheet someone dropped on you. In other words, you have become a server. On the other hand, you can pass on other people's owner links with impunity.

Once links are involved, the complexity of the model from the user's point of view grows. There are extra options, so that the user can decide when links are updated, and there is a mechanism for repairing broken links. It is worth remembering that all this sophistication sits on top of the DOS file system, which has no notion of links; and there is nothing to stop you from going to DOS and making a complete mess of the links by deleting files pointed to by links. Presumably NT-hosted versions of Windows will be more careful about this.

Microsoft advocates widespread use of OLE. You should be prepared to have almost anything pasted into your documents; indeed, there is a special gizmo called the packager that turns innocent pieces of bric-a-brac like files and command lines into packaged objects that can be embedded in compound documents; you are not even safe from non-OLE aware applications. The rationale for this is that the user should be in control of a task and should determine what tools are used; you should never assume that your application is all a user will need. OLE allows a user to organise files according to tasks, not applications. As an aside, it also allows the disorganised user an extra dimension of carelessness. Did I really drop last month's accounts in a chapter of my textbook? Whatever happened to that nice diagram I meant to send to .EXE that was teetering on the brink of the Prolog compiler? Please Mr Norton, can we have a 'where did I lose that object?' utility?

In Prolog-2, we allow a program file to have embedded objects; a programmer can thus keep all the pictures associated with a program in the program file. This is definitely useful. More speculatively, we allow bits of compiled Prolog to be embedded in other files; Prolog acts as a server to execute the code via an OLE verb. This might be useful if you wanted to store some logical rules in a spreadsheet to evaluate a cell, say; but this seems to me less obviously useful and it will be interesting to see what users make of it.

Drag and drop

The simplest way to OLE data from one application to another is via the clipboard; but other means are envisaged too. For example, a client application might have an option to create a new embedded option, or to link directly to a file chosen from a menu. For the user, a particularly simple approach is to pick up a file in the file manager and drop it on the client.

If you are concerned only to collect objects you understand, drag and drop is easy to implement. First you must call `DragAcceptFiles()`, telling Windows that you accept dropped files. The effect of this is visible to the user for if you pick up a file in the file manager and carry it around the screen then it turns into a no entry sign over most applications, but is shown as a small document marked with a plus over documents onto which it may be dropped. Groups of documents may be moved too.

When the user releases the mouse button over your application, the files are dropped onto you. You receive a message called `WM_DROPFILES` with a handle that can be interrogated to find out how many files were dropped and what they were called. You can then sort through them, process any with your applications extension, and ignore the rest. Incidentally, file extensions are an important part of the OLE and registration mechanism, because they enable the registration system to find the right application to process a file. If you capriciously give all your documents the extension .XLS things will not work well.

I have said that Microsoft advocates *unrestricted* OLEing of data between applications, so really you ought not to throw away those files that were dropped onto you. Nor are you supposed to find their owner in the registration database and send it a message asking it to come and take its rubbish away.

THE CONCEPT BEHIND OUR CASE PRODUCT.

System Architect not only has the power and flexibility to handle your most complex application, it is *easy to use* and *affordable*. As the only CASE tool to be specifically developed for MS Windows, System Architect brings you a graphics oriented, fully integrated multi-user development platform for PCs.

In addition to standard CASE tool features, System Architect offers a wealth of additional functionality.

Methodologies Supported

- SSADM
- Yourdon/DeMarco
- Ward & Mellor
- Object Oriented
- Gane & Sarson
- Information Engineering

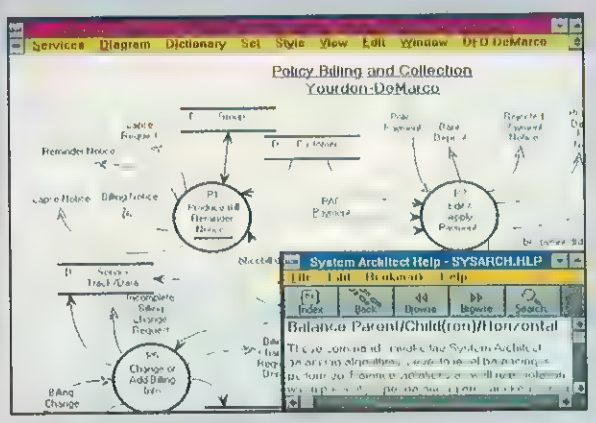
Full Data Modelling

Lifecycle Support

- Requirements Tracking
- Analysis/Design
- Prototyping
- Schema Generation
- Code Frame Generation

Benefits

- Ease of Use
- Readily Affordable
- Networking
- PC Based
- Increased Productivity



To arrange a demonstration, attend one of our regular System Architect seminars, or for further information please phone us on

Tel: (0926) 450858

Fax: (0926) 422165

Special Features

- User Definable Attributes
- Customisable Reporting System (Matrix and Tabular)
- Automated Documentation
- Import/Export/Extract/Merge
- Multiple Document Interface
- Import of External Graphics
- CRUD Matrices
- Mini-Spec Balancing
- OS/2 Support

Proven Reliability

- Over 6,500 licences installed. Users include:
  - NCR • Toshiba • Racal
  - IBM • Bull • Prudential

Training and Support

Full training and support for both the novice and advanced user. Also training courses in Structured Methods and Object Oriented approach.

MICROSOFT WINDOWS
Version 3.0 Compatible Product

Supporting IBM's AD/Cycle

Real-Techniques and Methods Limited
118-120 Warwick Street, Royal Leamington Spa,
Warwickshire, England CV32 4QY.

System Architect™

CIRCLE NO. 343

You should use the packager to package the file and embed the resulting package in whatever you happen to be working on.

File manager is the only source of drag/drop messages; you mustn't pick up bits of lost property from one application and move them to another.

Applets

Whatever happened to all those people who used to turn an honest penny writing filters to remove every fifth X from files? The Windows development environment has appealed mostly to writers of applications with lots of built-in functions, and Microsoft could hardly be said to have resisted this trend in their own products. It could be argued that once you have gone to the trouble of building a fancy Windows interface, you may as well put a lot of things on the menus. However, it is argued, OLE makes possible again the development of simple applications that do a single job well. They will be OLE servers, with very little user interface code, because they will not need their own file manipulation menus, for example; they exist only to serve data in other peoples' files. And indeed, Word for Windows 2.00 comes with a collection of applets of just this kind.

Help!

Microsoft has added some important functionality to the help system. One enhancement is bitmap graphics with hot spots, which makes explaining what a dialog box does much easier. When the user clicks a button or area on the picture of a dialog, a pop-up window will explain what the dialog does. Figure 2 shows the system in operation and also allows me to show you the new file open dialog, of which more later.

Applications can now add application specific-buttons to help, and the browse buttons do not necessarily appear. Thus, if you want to add the kind of help system you find in Excel, which can actually demo instructions being performed, you add an appropriate button and cause this to invoke code in a DLL that you supply. Code can also be attached to hot spots in the text.

In Prolog-2 we use an extra button to access a system for constructing a Prolog goal from a series of menus, and then allow the goal to be copied to the clipboard. This helps obviate the sense of frustration the user feels when the desired code is clearly printed in the help window but has to be laboriously retyped in the application.

The common dialogs

The common dialogs allow the programmer to construct dialog boxes for common tasks. The point is not to make life easier for the programmer; most of us have constructed dialogs for these jobs already and will now have to throw them away. But a common interface makes life easier for the user, who can assume that everybody's file open box behaves in the same way. The file open box is particularly easy to use, with drives separated out from directories and a graphic representation of ancestor directories (see Figure 2 again). The little combo at the bottom left allows the user to choose between files of different types and is, I suspect, a small concession to the OLE theory that any application should be prepared to open any file.

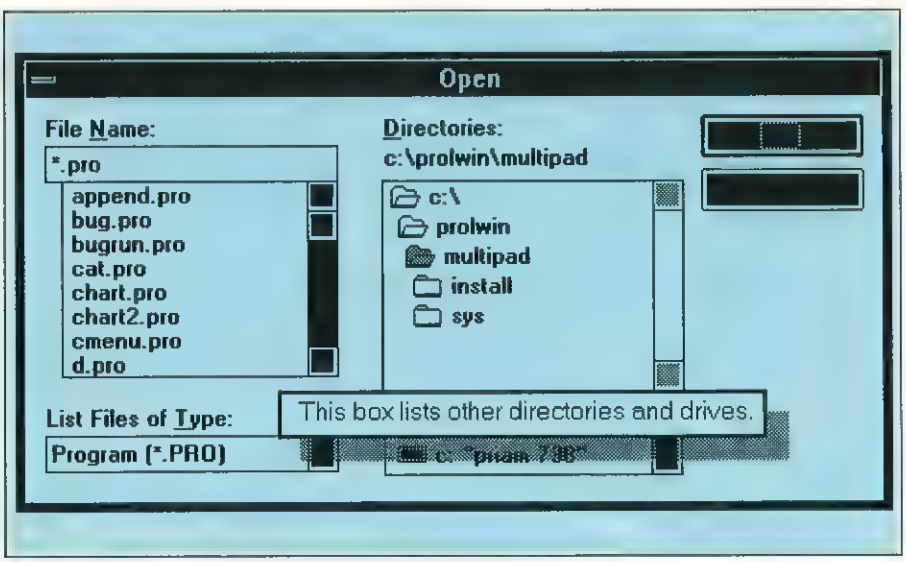


Figure 2 - Help screen for the standard file open dialog

Apparently Microsoft has discovered that 60% of UAEs result from passing incorrect parameters to Windows calls. So would you please stop doing it. Thank you.

No, hang on, there was something else. UAEs have been brightened up and made more fun. As a wacky new experiment, Microsoft are going to check the values of parameters passed into API calls. UAE messages will now point the finger at the guilty party in a message elegantly centred on a sombrely bordered background, rather like on a tombstone:

```
Tony Dodd

crashed this software

by passing an integer

to SetDlgItemText

EHEU!

PoorDeadApp will close
```

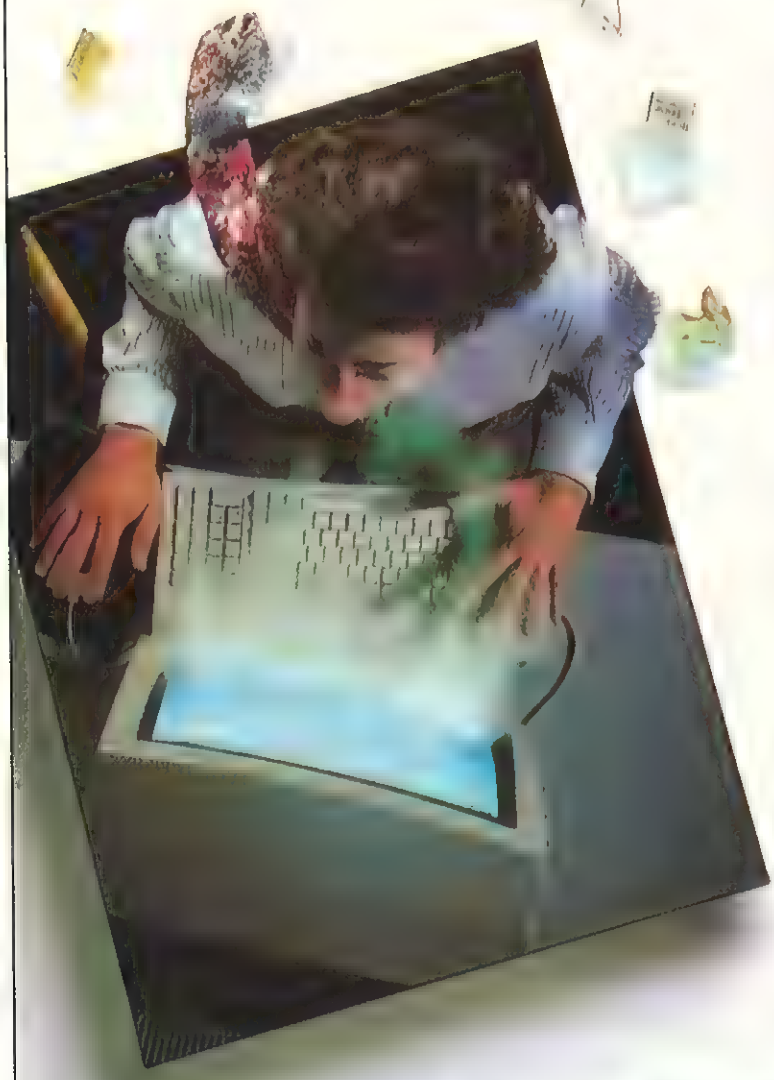
Unless I misunderstood the presentation at the developers' conference, it is intended that the user can choose to ignore this and carry on with the incorrect parameters, with the undead PoorDeadApp stomping around in memory.

Installation

Installation of Windows software is becoming a complex business, especially with developers shipping beta releases of Windows on their own disks. It becomes very important to make sure that a newer version of a DLL does not get overwritten by an older one. To this end Microsoft now ships a version control DLL, and has started version stamping all its files. There also seems to be a mechanism in the resource compiler to allow developers to stamp their own files. Moreover, the compression logic used in the EXPAND and COMPRESS programs is being made available as a DLL.

Then there is the question of OLE. The registration database associates files with applications using their extensions, and for each application it records information essential to the OLE mechanism. For example, somebody drops a file with extension .XLS on my application. I need not only to find out who owns the data (that information is probably available from the extensions section of WIN.INI) but also to find what command line must be used to awaken the owner, what actions the owner can perform; and the OLE libraries, which manipulate OLE data using DDE, need still more recondite information. All of this is

Unleash The True Power Of C++



It Takes More Than The Language.

Unleashing the full potential of C + + takes powerful object-oriented tools and a rich library of classes. Using only a C + + compiler and traditional tools yields traditional results – slipped schedules and maintenance nightmares. Only Objectworks\C + +, Release 2.4, provides the comprehensive development environment that allows you to produce extraordinary results.

Speed Your Development and Maintenance.

Objectworks\C + + provides dynamic, graphical browsers to illuminate the tangle of class relationships, object interactions, and program structure, as no static or textual information possibly can. The unique integration of the C + + source level debugger and interactive browsers ensure that essential information is instantly at your finger tips. And, a broad range of cross references are as close as a point and click. All of these tools aid in the understanding of code, increase reuse, decrease confusion, promote consistency and, ultimately, speed your development and maintenance.

An Open Environment.

Based in the latest AT&T C + + language System, Release 2.1, Objectworks\C + + provides support for SunView and X Windows on Sun platforms. Its open environment also allows you to use your favourite C preprocessor, C compiler, linker, profiler, or source code control system. The sophisticated class libraries included in Objectkit\C + + exploit the natural synergy between the development environment and the extensive reusable class libraries. Objectworks\C + + also allows programmers to use existing makefiles without any modification, saving time and money.

Team Programming.

Objectworks\C + + is the only C + + development environment that allows engineers to work cooperatively, loading and browsing one another's code, without interrupting each other. Programmers simply access an information file containing a description of the code they need, and then they query this description.

AI International are uniquely qualified to help you exploit the benefits of object-oriented technology and C + + programming applicable to your development efforts. If you'd like our technical management brief "Exploiting the Full Benefits of C + + Objects", or are ready to unleash the true power of C + +, call us today at 0442 876448

Without Objectworks, you're without objects.

AI INTERNATIONAL LIMITED

The Chapel, Park View House, 1 Park View Road, Berkhamsted, Herts, HP4 3EY
Telephone: 0442 876448 Fax: 0442 877997

CIRCLE NO. 344

Developing software on a LAN is fast, efficient, and dangerous.

We make it safe, controllable and auditable.

Experienced managers are rightly concerned about the integrity of LAN development environments. It's too easy to overwrite or corrupt files.

To safely develop on a LAN you need control over file changes, the ability to recover any module revision or system version, audit trails, and security that doesn't hamper productivity.

The solution: our LAN development environment, three tools that operate independently or snap together synergistically.

The primary element is PVCS Version Manager which provides version control and change management features, files and function security, automatic audit trail and detailed project reports.

The best way to access PVCS Version Manager is via our PVCS Professional Editor, an extensible control centre that simplifies access to any tool and provides industry strength editing. PVCS Professional Editor invokes PVCS Version Manager automatically. Just open modules to edit and the PVCS Version Manager window pops open.

The final element in our LAN development environment is PVCS Configuration Builder - a sophisticated tool for automatically performing correct builds. A single command recompiles all modules that have been changed, re-links as necessary, and produces a correct, conforming program.

Don't Let These Special Offers Pass You By.

Special offer prices are valid until 31st January 1992.

	DOS or OS2	DOS and OS2
PVCS Version Manager	£320	£416
PVCS Configuration Builder	£136	£180
PVCS Professional Editor	£90	£115
Mainframe Gateway	£112	—

Software Express · Portmill House, Portmill Lane, Hitchin, Herts SG5 1DJ. Tel: 0462 422525

CIRCLE NO. 345

stored in the registration database. If you ship an OLE-aware application you need to produce the appropriate entries for your software and merge them with the existing database on a machine as part of the installation procedure.

And more...

A number of new and altered tools are now supplied. There is a thing called Dr Watson - whoever designed the icon can never have seen Nigel Bruce in The Hound of the Baskervilles - that lurks in the background taking notes of UAEs, for whose use it is not clear. There is a stress application that tests how software behaves as various resources grow scarce; it is considerably crueller than the heapwalker in its battering of applications.

Then there is a whole new bunch of DDE calls to make DDE easier (26 new function calls, 9 new structures, and 142 new defined constants according to Petzold). One more round of making it easier and it will be completely impossible.

If software suppliers are allowed to ship all the Windows 3.1 DLLs with their software, and if these will work with Windows 3.0, why will users ever bother to upgrade to 3.1? There are several good reasons. The new file manager, essential for drag and drop, is part of retail Windows and not a DLL. The promised more robust handling of UAEs is exclusively in 3.1. And there are a whole number of small fixes and improvements, which developers will be unable to resist using. I suspect that most developers will freeze their 3.0 product and make new releases dependent on 3.1.

In the long term, it is intended that there should be a new mode for Windows. At the bottom end, real mode has been lost, and standard mode is now the low end. The new top end mode will be NT mode, based on Microsoft's new kernel NT operating system, and will offer security features and support for a RISC processor. Windows 32 is a new API that will appear in Windows in 386 enhanced and NT modes; as well as 32-bit addressing and a general clearing away of 64 KB limits, this will offer preemptive multi-tasking with threads and better graphics, including Bezier curves. The first appearance of the Windows 32 API will be in the NT-hosted Windows product; this is already in the early stages of testing. For most developers the Windows 32 API, especially when it appears in 386 enhanced mode under DOS, will be a far more significant challenge than Windows 3.1. Windows 3.1 contains some extra features, such as more handle types and type-checked versions of message functions - designed to make the transition to Windows 32 easier.

Thus, you should stop assuming that all handles can be typed `HANDLE` and start using `HPEN`, `HMENU` etc, because in Windows 32 different types of handle may have different widths. And you should stop passing any old garbage around as message parameters and using casts to persuade the compiler to look the other way. In the long term it may be these internal changes in 3.1 that are of most benefit to Windows developers.

EXE

Tony Dodd is Technical Director of Expert Systems Ltd and currently holds an SERC/Royal Society Industrial Fellowship at Bristol University Computer Science Department. He developed Expert Systems Ltd's Prolog-2 for Windows 3 and recently upgraded the product to take advantage of the new features of Windows 3.1

Keeping up with the Kahns

Borland C++ version 3.0 is the company's first optimising C++ compiler and comes complete with the long-awaited Turbo C++ for Windows. Paul Kemp wheels out the benchmarks.

These days it seems that barely a week goes by without yet another new product announcement from Borland. While Microsoft dithers about with C version 7.0 (its C/C++ compiler), Borland is releasing a third-generation professional C++ compiler and a Windows-hosted C++ environment for developing Windows applications.

The boxes

The three packages that are affected by the release of Borland C++ 3.0 (BC30) and Turbo C++ for Windows (TCW) are listed in Figures 1-3. The first two bundles are upgrades of existing products, while Turbo C++ for Windows is an entirely new product. For the purposes of this article, I was using the comprehensive Borland C++ 3.0 & Applications Frameworks (Resource Workshop, Turbo Vision for C++ and the ObjectWindows library (OWL) were reviewed in November's issue of *.EXE*). Clearly you get a lot for your money.

BC30 & Application Frameworks arrives on 17 diskettes and takes about 35 minutes of intensive disk-juggling to install. It also gobbles up nearly 29 MB of disk space. Unfortunately, the new manual set was still being printed at the time of writing, but I was assured that it would be more complete than the documentation that accompanied the initial release of Application Frameworks. The documentation for BC30 should include the following:

- Borland C++ User Guide
- Borland C++ Tools & Utilities
- Borland C++ Programmer's Guide
- Borland C++ Library Reference
- Turbo Debugger User's Guide
- Turbo Assembler User's Guide
- Resource Workshop User's Guide
- Windows API Reference

ANSI C and C++ v2.1
Global optimising compiler
DPMI compiler and text-mode IDE
Turbo C++ for Windows with Object Browser
WinSight utility for tracking Windows messages (Windows)
Turbo Debugger for DOS and Windows (text-mode)
Turbo Profiler for DOS and Windows (text-mode)
Turbo Assembler
Resource Workshop

Figure 1 - Borland C++ 3.0

[as for Borland C++ 3.0 PLUS]
ObjectWindows application framework for Windows (including source code)
Turbo Vision application framework for DOS (including source code)
Run-time library (RTL) source code

Figure 2 - Borland C++ 3.0 & Application Frameworks

ANSI C and C++ v2.1
Windows-hosted IDE
ObjectWindows application framework for Windows
Object Browser graphical source browser
EasyWin library for porting DOS programs to Windows
Turbo Debugger for Windows
Resource Workshop

Figure 3 - Turbo C++ for Windows

What's new

The BC30 compiler is compliant with AT&T's C++ v2.1 language specification, but includes support for v3.0 templates (*.EXE* November '91 *News* section). Version 2.1 of C++ implements some minor modifications to the language. For example, in C++ 2.0, when an array of objects is deleted using the `delete` operator, the size of the array must be specified:

```
obj * pobj = new obj[5];
.
.
delete[5] pobj;
```

With C++ 2.1, the array size no longer needs to be (and, in fact, may not be) specified with the `delete` operator:

```
obj * pobj = new obj[5];
.
.
delete[] pobj;
```

However, in order to allow v2.0 code to compile, BC30 issues a warning and simply ignores any size that is specified.

BC30's command-line compiler, linker and text-mode integrated development environment (IDE) are all hosted under DPMI (DOS Protected Mode Interface - *.EXE* April '91). This means that the DOS-extended

Windows Developers!

Question: ***I need Images & Special Effects into my Windows application?***

Answer: WinImage SDK!

Your Windows application deserves the best. And that's just what **WinImage** delivers!

Escape the Code Trap.

Eliminate the time-consuming process of learning and writing code necessary to support different types of images. **WinImage** fully supports each type of image format.

The Object of Desire.

WinImage is an easy to use DLL and is very easy to customize. Complete with a full set of documentation to allow the programmer to create complete, beautiful, robust applications in record time and is **Royalty Free!**

Supports all Windows compilers. Includes support for TIFF, PCX, Encapsulated Postscript, Bitmaps, GIFF, Metafiles, Numerous Dazzling Special Effects F\X, Animation and much more.

Price: 395 (with Source 995)

Call now for Special Introductory offer.

PictureBox F\X

From the people that brought you the World's finest presentation and low cost MultiMedia presentation package specializing in Real Corporate Presentation, now brings you the same Special Effects again but for Windows. Buy now and Save! We promise that you will not be disappointed! **For Only: 395**

Call now for Very Special Introductory offer.

HIGHLAND *Grafix*

18 Albion Way, East Kilbride
Scotland G75 0YN (03552) 64888 Sales
(03552) 64777 Fax

All Specifications subject to change without notice.

CIRCLE NO. 347

JPI

the compiler people

Imagine a powerful integrated development environment common to whichever languages you choose, capable of supporting DOS, Windows or OS/2 development embodying the latest in OOP technology – it's called ***TopSpeed*** from JPI. Its modular architecture opens up new realms of choice – simply slot-in languages, source libraries, and toolkits to build a development system that meets your needs. No redundant components – what you want is what you get!

***TopSpeed* Environment** – Multi-window editor, powerful Project system, Hypertext help (environment, all languages and library), debugger, syntax checker, menu or command line driven, EMS support, pop-up calc, plus . . . plus . . . DOS or OS/2 **£59**

***TopSpeed* TechKit** – For power programming. Supports DOS DLLs, post mortem debugging, advanced overlay manager, TopSpeed assembler, .EXE file compressor, .OBJ file disassembler. **Windows 3 resource compiler**, Windows 3 run-time libraries, plus . . . plus . . . **£59**

***TopSpeed* C++** – makes C++ lean and mean Unique to JPI – TopSpeed C is the only true AT&T 2.1 C++. TopSpeed C++ gives you SmartMethod® Linking which eliminates unreferenced classes, methods and even virtual methods – a real breakthrough for OOP programming. Includes short based pointers, concurrent tasking – even with DOS, plus . . . plus . . . DOS or OS/2 **£59**

Rogue C++ Class Library **£59** DOS or OS/2

***TopSpeed* C** – *the standard is enhanced.* The only ANSI certified C. Generates compact high quality code for DOS, OS/2 and Windows 3. Includes run-time error checking, multi-threading, mixed memory models, links to C++, Modula-2 and Pascal, plus . . . DOS or OS/2 **£59**

***TopSpeed* Modula-2** – the world leading implementation of Modula-2. This strongly typed and highly structured language includes type safe conversion between objects, OOP extensions with true multiple inheritance, virtual pointers, based pointers, links to C, C++ and Pascal, plus . . . DOS or OS/2 **£59**

***TopSpeed* Pascal** – the next generation. Power-up your Pascal, convert to TopSpeed, then tune it up with the hottest compiler for DOS, Extended Dos, OS/2, or Windows. ISO 7185 conformant, Turbo to TopSpeed converter, ISO conformant arrays, dynamic strings, separate compilation units, OOP extensions plus much more . . . DOS or OS/2 **£59**

Library Source Kits available in all languages **£59**

NEW ***TopSpeed* DOS Extender**

Power-up your programming with JPI's own DOS Extender toolkit. Blast the 640K DOS barrier and supercharge your applications. Built-in disk based virtual memory management system for code *and* data, multi-tasking support using OS/2-like threads and a pre-emptive scheduler, OS/2 format DLLs supported, automatically loaded and unloaded on demand, plus . . . Royalty free. Full source code available.

Call JPI on (0234) 267500 now for your free copy of the TopSpeed 1991 Compiler Catalogue

TopSpeed products are available from:
GreyMatter 0364-53499 · System Science 071-833-1022
RTA 081-656-7333

JPI CIRCLE NO. 348

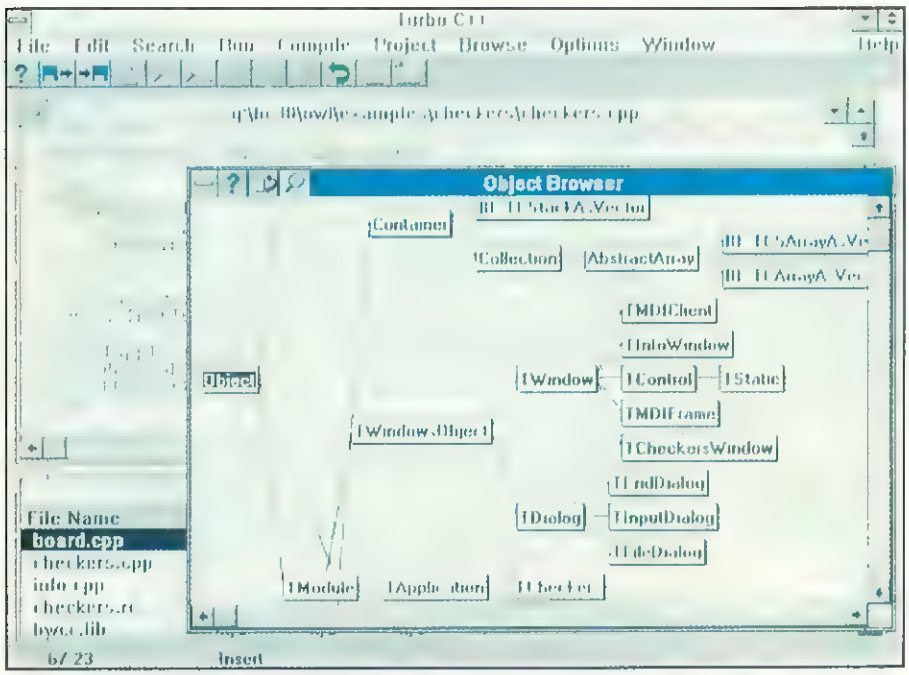
3 The Mansards, Tavistock Street, Bedford MK40 2RX
Fax: (0234) 217094

TopSpeed C++ TopSpeed Modula-2 TopSpeed Pascal TopSpeed C

versions of these utilities (BCCX, TLINKX and BCX) have disappeared and the new versions of BCC.EXE, TLINK.EXE and BC.EXE will take advantage of all available memory on the host machine (without having to unload your own EMS driver - such as DOS 5's EMM386.EXE). This is a distinct improvement over V2.0, obviating the need to modify CONFIG.SYS and reboot when using extended memory.

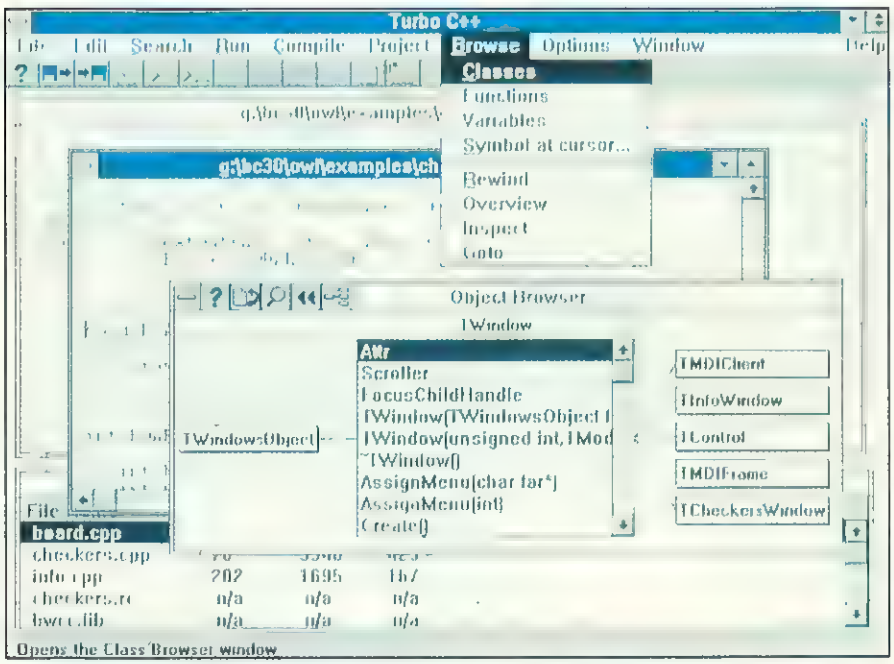
Turbo Debugger has been improved to support debugging of C++ v2.1 features such as nested classes, as well as templates and debugging of optimised code. It returns correct values for variables enregistered by the optimiser, and will not return values for variables eliminated by the optimiser. There is also now a clipboard (similar to the one in the IDE) that lets you cut and paste addresses, data, code and text from window to window. Improvements have been made in the ability to debug remote applications (via a NetBIOS-compatible network as well as through serial port connection) and breakpoints can now be set with multiple conditions and actions. In addition, Turbo Debugger for Windows has been enhanced to enable the simultaneous loading of multiple symbol tables, making it a lot faster to debug DLL code. There is also a new Selector pane in the CPU window which displays whether a selector is a code or data selector; the size in bytes of the memory segment it references; whether the segment is currently loaded into memory; and, if the selector is a data selector, whether it expands 'upwards' or 'downwards' in memory. Unlike the debugger in QuickC for Windows, Turbo Debugger for Windows is still a character-based application, even though it is a Windows program.

It will be interesting to see whether assembler-junkies will approve of the OOP invasion



ADVANCED BUSINESS GRAPHICS

Figure 4 - Turbo C++ for Windows' Object Browser



ADVANCED BUSINESS GRAPHICS

Figure 5 - Data & methods displayed in TCW's Object Browser

I admit to being relatively inexperienced in the use of code profilers, but even so, I did find Borland's Turbo Profiler somewhat baffling. Perhaps it is because there are so many features packed in. The new version of Turbo Profiler has a number of extra facilities over its predecessor. Profiling of Windows programs, in all modes, including local, remote serial and remote network, is now supported and brings it on par with the Microsoft equivalent. Coverage analysis detects sections of code that don't get executed when a program is run; helping programmers to ensure that all parts of the code are tested. It is also possible to automate profiling using DOS batch files, and save statistics in a .TFS stats file.

It seems that no piece of software is safe from Borland's OOP crusade, even to the extent that Turbo Assembler 3.0 (TASM 3.0) has object-oriented extensions. An object in TASM 3.0 has both a table of virtual methods and a data structure associated with it. The object layout is similar to that of objects in Turbo Pascal and is defined by the following syntax:

```
<name> STRUC {<modifiers>}
{<parent_name>}
{METHOD <method_list>}
<structure_data>
```

Only single inheritance is explicitly supported and the OOP extensions do not support mixed-language programming - TASM 3.0 objects are not link-compatible with BC++ or Turbo Pascal. It will be interesting to see whether assembler-junkies will approve of the OOP invasion of this last bastion of procedural programming.

Asynchronous Point-to-Point Communications or Telephone-polling Multiple Computers

XOREN COMMS SOFTWARE– Software with ten years' pedigree

Thoughout the 1980s, Xoren Computing pioneered the development of inexpensive yet reliable communications software. Today, the XOREN IPL-11 range is renowned for enabling automated file transfer between any quantity and combination of the major minis, micros and PCs in configurations which will become more and more commonplace through the 1990s.

Leading with Automated Connectivity

Although many other communications packages can asynchronously link PCs together, very few besides IPL-11 will orchestrate *fully automated connectivity* between multiple PCs and hosts.

Automated operation with IPL-11 gives users the freedom to perform unattended data communications 24 hours a day. Computers with modems operating up to 9600 baud can be scheduled to dial up remote computers over ordinary telephone lines day or night to send and receive files. Any IPL-11 version can link and transfer files to any other IPL-11 version, regardless of the computers' operating system. Once the files have arrived safely, telephone lines are disengaged.

Leading with Customer Support

Key to the success of the IPL-11 range is an emphasis by Xoren on customer support.

Because communications is notorious for being a difficult application to implement, Xoren offers installation and support services to ensure the software can operate according to the *user's* requirements.

This principle has made IPL-11 communications software the obvious choice for UK and European companies. Because IPL-11 is a British product, developed and supported in the UK, Xoren is strategically placed to support customers when they need support most - during regular business hours.

Call or write to Xoren today - the most practical solution to automated data communications has always been right on your doorstep.

XOREN IPL-11 Developments over a Decade

1980: Xoren launches IPL-11. Features include peer-to-peer file transfer under operator control, CRC error-checking on individual packets and queuing of file transfer commands
Original versions introduced for RSX-11M and VMS for DEC PDP-11 and VAX.

1981: Remote activation facility incorporated allowing file transfers when remote computers are unmanned.
RSTS/E, RT-11 and RSX-11M PLUS versions released for DEC PDP-11.

1982: Terminal emulation facility introduced enabling the use of a terminal on a local computer as a terminal on a remote computer thereby allowing control of file transfer sessions from a single terminal.
TSX PLUS version released for DEC PDP-11.

1983: Option to control file transfers from command files as an alternative to control from operator's keyboard
P/OS version released for DEC Professional.

1984: Mechanism built into the package protecting against "message bouncing" due to line noise when computers remain connected and the package is not in use.
MicroRSX and MicroRSTS versions released for DEC MicroPDP-11.

1985: Commenced development of new portable versions written in the programme language 'C'.

1986: First releases of new portable versions written in 'C' for PC-DOS, MS-DOS, UNIX, AIX and VMS.

1987: Portable versions support simultaneous multiple links.
IPL-11 wins ICP Million Dollar Award.

1988: PC versions enhanced with improved terminal emulation including VT100 emulation, keyboard mapping and facilities to define function keys.

1989: Comprehensive upgrade for unattended operation of multiple PC/host links supporting auto-dialling modems.

1990: **Release of MULTI-POLL range of polling software incorporating IPL-11 allowing PCs or other computers at multiple sites to be telephone-polled by a central host computer system to transfer data to and from the central system.**

XOREN COMPUTING LTD
28 MADDOX STREET
LONDON W1R 9PF
TELEPHONE 071-629 5932
FAX 071-629 5432

I am interested in Xoren Comms Software:

Name _____

Position _____

Company _____

Address _____

_____

Postcode _____

.EXE 12/91

Optimise!

As already mentioned, BC30 is Borland's first C++ compiler to include a global optimiser. An optimiser attempts to generate more efficient machine code by eliminating unnecessary operations and speeding up instructions by whatever means are available. The optimisation switches that are available in BC30 are listed in Figure 6.

Some speed optimisations actually make the code larger but faster. The -Oi switch, for example, expands the functions listed in Figure 7 to inline code, thus removing the overhead of a function call but potentially enlarging the code size.

In order to test BC30's performance against other popular C compilers, I used a set of simple benchmarks designed by Dr Thomas Plum, vice-chair of the ANSI C X3J11 committee. The programs output a table of six figures that give an indication of how many microseconds it takes to execute the 'average operator' on `register ints`, `auto shorts`, `auto longs`, and on `double` data, as well as the time for an integer multiply and the time to call-and-return from a function. The optimisation switches used for the compilers tested are in Figure 8 and the results (for a 25 MHz/386 machine) are tabled in Figure 9.

It can be seen that BC30 did not actually come out on top in any of the categories. However, in compile and link time, it was not far behind the super-swift Zortech compiler. There is one thing worth noting though, because of a bug in the beta version of the optimiser associated with BC30's -Og switch, it was necessary to disable this particular type of optimisation in the test. Although the results show that Borland's optimiser does improve code efficiency in many of the tests, it does not appear to be as effective as the other compilers that were looked at. Figures 10 and 11 show data published by Borland for other popular benchmarks. These tests show the compiler in a more favourable light compared to Microsoft C 6.0 (natch).

I carried out two further very simple benchmarks (also suggested by Borland) on this selection of compilers. The first, ALLOCMEM.C (Figure 12), allocates and frees a 10 KB block of memory 50,000 times. The second, NEWALLOC.C (Figure 13), performs this memory allocation only 5,000 times but also allocates one byte of memory, which is not freed immediately, each time round the loop. This can be considered a more 'realistic' benchmark than ALLOCMEM, since in real world programs, memory is not always freed in the same order as it was allocated. The results of these two test are presented in Figures 14 and 15. While MSC6 performed best at ALLOCMEM, it was staggeringly slothful when executing NEWALLOC - Microsoft technos take heed. The results for BC30 showed it to be comparable to (but not better than) the TopSpeed and Zortech compilers.

In addition to the standard C-type optimisations discussed above, BC30 also imple-

Switch	Explanation
-O2	Generate the fastest code
-O1	Generate the smallest code
-O	Remove unnecessary jumps and unreachable code
-Oa	Assume pointer expressions are not aliased in common subexpression evaluation
-Ob	Eliminate stores in dead variables
-Oc	Enable common subexpression elimination within basic blocks only
-Od	Disable all optimisations
-Oe	Enable global register allocation and variable live range analysis
-Og	Enable common subexpression elimination within an entire function
-Oi	Enable inlining of intrinsic functions
-Ol	Compact loops into REP/STOSx instructions
-Om	Move invariant code out of loops
-Op	Propagate copies of constants, variables and expressions
-Os	Make code selection choices in favour of smaller code
-Ot	Select code in favour of higher speed
-Ov	Enable loop induction variable and strength reduction optimisations
-Ox	Enable most speed optimisations (provided for Microsoft compatibility)
-Z	Suppress reloads of enregistered values
-pr	Enable the _fastcall calling convention for passing parameters in registers

Figure 6 - Optimisation switches in Borland C++ 3.0

memchr	memcmp	memcpy	memset	stpcpy	strcat
strchr	strcmp	strcpy	strlen	strncat	strncmp
strncpy	strnset	strrchr	rotl	rotr	fabs
alloca					

Figure 7 - List of intrinsic functions that may be inlined

Compiler	Speed optimisation switches
Borland C++ 2.0	-G
Borland C++ 3.0	-O2-g
Microsoft C 6.0	/Oazx
TopSpeed C 3.02	(all optimisations turned on by default)
Zortech C++ 3.0	-o

Figure 8 - Optimisation switches used with different compilers

Compiler	register int (microsec) -opt	+opt	auto short (microsec) -opt	+opt	auto long (microsec) -opt	+opt	int multiply (microsec) -opt	+opt	function call + ret (microsec) -opt	+opt	auto double (microsec) -opt	+opt	.EXE size (bytes) -opt	+opt	compile & link time (seconds) -opt	+opt
Borland C++ 2.0	0.45	0.43	0.45	0.45	1.59	1.57	0.99	0.95	2.51	2.47	147	147	27704	27720	9.44	(9.44)
Borland C++ 3.0	0.45	0.25	0.45	0.25	1.56	1.46	1.25	1.20	2.51	1.76	147	147	28114	28050	16.70	17.68
Microsoft C 6.0	**0.33**	0.38	0.44	0.38	**1.10**	1.10	**0.33**	**0.27**	5.00	**1.60**	66	60	32454	32422	28.0 0	40.54
TopSpeed C 3.02	0.34	**0.23**	**0.34**	**0.23**	1.20	**0.88**	1.21	1.15	2.53	1.86	76	56	**26384**	**25964**	26.09	26 80
Zortech C++ 3.0	0.41	0 36	0.55	0.37	**1.10**	1.15	1.20	0.95	**1.76**	1.76	**24**	**24**	36722	36562	**9.17**	16.43

NOTE: +opt data for speed-optimised code only. Faster times and smaller sizes are in bold face.

Figure 9 - Plum's benchmark results

Benchmark	Execution speed MSC6	Execution speed BC++3	Code size MSC6	Code size BC++3
SIEVE	18.40	13.25	161	**105**
DRYSTN	10.17	9.91	1440	**1189**

NOTE: Data supplied by Borland International. Performed on 386/33 with 4 MB disk cache. Faster times and smaller sizes in bold face. Code sizes and execution speeds are for size and speed-optimised code, respectively.

Figure 10 - Sieve and Dhrystone benchmarks

Benchmark	Microsoft C 6.0	Borland C++ 3.0
Compile/link time	187.6	56.1
Execution speed	30	28
Code size	84030	78613

NOTE: Data supplied by Borland International. Performed on a Dell 433e (486/33) with 4 MB disk cache. Faster times and smaller sizes in bold face. Code sizes and execution speeds are for size and speed-optimised code, respectively.

Figure 11 - XSCHEME results

ments several optimisations that are transparent to the programmer. These include virtual function and base optimisations - when calling virtual functions or accessing virtual base classes, whenever the compiler is able to determine the true type of the object being accessed, it binds the function or base class address at compile time. Using thunks for virtual member pointers - for pointers to function members of classes, instead of encoding explicity whether the member being pointed to is a virtual function, the information is implicity encoded through the use of virtual call thunks. Passing classes by value - when an argument of type class with user-defined constructors is passed by value to a function, older versions of BC++ would pass a reference to a temporary instead. BC30 will copy-construct the argument value to the stack.

```
/* ALLOCMEM.C */
void * malloc( unsigned int );
void   free( void * );

main()
{
  char        * x;
  unsigned int i;

  x = malloc( 10000 );

  for( i = 0; i < 50000; i++ )
  {
    x = malloc( 10000 );
    free( x );
  }
}
```

Figure 12 - ALLOCMEM.C

Turbo C++ for Windows

When I reviewed Microsoft's QuickC for Windows in October, I was bemoaning the fact that there wasn't a Windows-hosted C++ compiler. Well, now there is one. Turbo C++ for Windows is Borland's C++ equivalent to Turbo Pascal for Windows, and can be used to develop Windows applications (not DOS) using the company's ObjectWindows library. The menu options, dialog boxes and child windows are all very similar to the text-mode IDE, so it is easy for anyone accustomed to that environment to make the switch. It also sports the now ubiquitous row of cryptic icons under the main menu (coined a *SpeedBar* by Borland, since the name *ToolBar* is copyrighted by Microsoft). Once I had managed to decipher the symbols, I did find it very useful, especially as it is context-sensitive and transforms itself depending on the active window. But the main enhancement that TCW offers, other than being Windows-hosted, is the Object Browser.

Figures 4 and 5 show TCW's Object Browser in action. It is invoked by the Browse menu (Figure 5) or by clicking the right mouse button on an item of class data or a method in a section of source code. One can view a graphical representation of the class hierarchies used in an entire application (Figure 4), zoom in to view data and methods (Figure 5) and finally probe right down to a method's implementation in a C++ source file. This kind of functionality is quite unlike anything currently available on the DOS/Windows platform, and is approaching the sophistication of C++ programming environments that are available on UNIX (such as ParcPlace's ObjectWorks or Saber C++). A couple of criticisms are that scrolling the graphical hierarchy display is rather slow and improvements could be made in the way methods and class data are shown (for example, there is no indication of whether a member is `public`, `private` or `protected`). However this utility is a real boon for C++ programmers and should prove to be immensely useful.

```
/* NEWALLOC.C */
void * malloc( unsigned int );
void   free( void * );

main()
{
  char        * x;
  char        * y;
  unsigned int i;

  x = malloc( 10000 );

  for( i = 0; i < 5000; i++ )
  {
    x = malloc( 10000 );
    y = malloc( 1 );
    free( x );
  }
}
```

Figure 13 - NEWALLOC.C

Compiler	run time (secs) -opt	run time (secs) +opt	.EXE size (bytes) -opt	.EXE size (bytes) +opt	compile & link time (secs) -opt	compile & link time (secs) +opt
Borland C++ 2.0	1.81	1.76	3850	3850	4.61	4.62
Borland C++ 3.0	1.75	1.76	4092	4092	9.29	9.78
Microsoft C 6.0	**1.21**	**1.27**	2907	2907	6.15	6.31
TopSpeed C 3.02	1.87	1.81	1882	1866	**14.61**	**14.56**
Zortech C++ 3.0	1.48	1.48	**1828**	**1812**	**2.97**	**3.73**

NOTE:+opt data for speed-optimised code only. Faster times and smaller sizes are in bold face.

Figure 14- ALLOCMEM program results

"The Best low cost alternative personal computer"
P.C. World

Atomstyle 386 SX came top in group review.
P.C. Plus June '91

On the APC1 486/33, *"You can't get a system much faster than that, and it's just the job for a programmer with a slow compiler"*
P.C. Plus

WINTER COLLECTION

286 16MHz	386SX 20/25 MHz	386 25MHz	386 33MHz	386 40MHz	486 33MHz
(SY 228) Expandable to 4MB on board.0 wait state. Landmark speed 21MHz. Expansion slots: 6x16 bit, 1x8 bit.	*(SY 366)* Expandable to 8MB on board.0 wait state. Landmark speed 26MHz. Expansion slots: 6x16 bit. Call for Slim Case option.	*(SY 334)* Expandable to 8MB on board.0 wait state. Landmark speed 33MHz. Expansion slots: 6x16 bit, 2x8 bit.	*(SY 374)* **64K CACHE MEMORY** Expandable to 32MB on board + EMS. Landmark speed 58.7MHz. Expansion slots: 7x16 bit, 1x8 bit.	*(SY 380)* **64K CACHE MEMORY** Expandable to 32MB on board. 0 wait state. Landmark speed 70MHz. Expansion slots: 6x16 bit, 2x8 bit.	*(SY 433)* **64K CACHE MEMORY** Expandable to 32MB on board. 0 wait state Landmark speed 156MHz Expansion slots 6x16 bit, 2x8 bit.
£570	**£660** £4.68 PER WEEK	**£715** £5.11 PER WEEK	**£820** £5.72 PER WEEK	**£865** £6.01 PER WEEK	**£1,170** £7.97 PER WEEK

SYSTEM SPECIFICATION *1MB Memory (4MB on 486) * 40MB Hard Disk * 14" MVGA Monitor 800 x 600 * 3½" 1.44MB Floppy * 102 Keys Keyboard * 1 Parallel/2 Serial * Desk Top Case * Built by highly trained technicians * All systems easily upgraded and all systems can be configured to customers specifications.

FREE
The following are supplied FREE with each system purchased
*1 YEAR on-site maintenance
(Subject to registration card return)
* DOS/MOUSE
* DUST COVERS

OPTIONAL EXTRAS Upgrade your 286/386/486 with these ***SUPER VALUE add-ons.***
(Please add to price of system)

SCREEN TYPE
HI Res VGA 800 x 600 VGA Card 256K £110
Super VGA Res. 1024x768 256 colours VGA Card 1MB £165

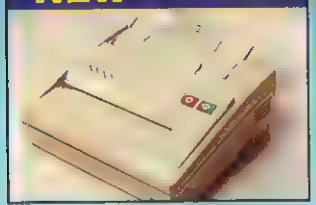
HARD DISK
105MB IDE 18m/s £100
180MB IDE 18m/s £220
330MB SCSI 17m/s £680
520MB SCSI 16m/s £805
IDE Cache Controller 1-4MB £215
1.2GB SCSI 13.5m/s £1215
SOFTWARE MS DOS 5.0 £25

MEMORY 1MB (extra) £35

FLOPPY DRIVE
5¼" 1.44MB £35

CASES
Mini Tower £25
Tower £55

APC 7
* Fax/Scanner
* Printer * Copier
The ideal low cost machine for the small business
NEW £425



NOTEBOOK COMPUTER
*** 386SX 20MHz * 60MB HD**
* 1.44MB Floppy
* **2MB** Memory
* VGA Mono 640 x 480
* 32 Gray scales
* RS232/Parallel/ Mouse/Scanner
* Rechargeable/ removable NiCAD battery 2.5hrs.

NEW £1,285

NETWORK SERVICE
In-House Network Technical Expert
NOVELL
RED BOX TD AUTHORISED DEALER

NEW SERVICE
Fully Tax Deductable ***LEASING*** *for companies*

START UP PACK
Installation plus ONE HOUR TUITION by highly qualified computer technician **£35**

LONDON
TEL: 081 801 1838
FAX: 081 365 1110
Atomstyle Limited, Unit 15, Lockwood Ind.Park, Millmead Road, Tottenham, London N17 9QU

LEEDS
TEL: 0532 442767
FAX: 0532 451980
Atomstyle Limited
Unit 21a Baker House,
9 New York Road, Leeds LS2 9PF

CAMBRIDGE
TEL: 0487 823908

Established 1982
HARD TO BEAT FOR PRICE & SERVICE!

APC ATOMSTYLE

CIRCLE NO. 351

YES Send this coupon back FREEPOST

Please call ☐
Please send me data on the following
286/386/486 Systems ☐ Notebook computer ☐
Network Services ☐ Workstations ☐ APC7 ☐
Laser/Matrix Printers ☐ Peripherals ☐
Please use BLOCK CAPITALS
Name_____
Company_____
Address_____
_____
Post Code_____ Tel_____
Please add my name to your mailing list ☐
.EXE 12/91

The debugging information embedded in a .EXE file is sufficient for Turbo Debugger for Windows and Object Browser to construct class hierarchies, but in order for Object Browser to link methods and data to source code, TCW's compiler needs to locate extra browse information in .OBJ files. Figure 16 shows compile times and .EXE sizes for the CHECKERS example program, supplied with OWL. It can be seen that the program actually compiled fastest in TCW and slowest when compiled using the BC30 IDE running in a DOS box under Windows. The results also show that the inclusion of browse information only added about 2% to the size of the executable.

Conclusion

Okay, Borland C++ has finally come of age and has a shiny new global optimiser. It might not be quite as aggressive as those available with other compilers, but then again it didn't seem to stop people buying the product even when it didn't have one at all. But that's not the whole story, there are many improvements and new features packed into this release which make it a significant upgrade. C++ v2.1 and template support; the DPMI-hosted IDE, compiler and linker; improvements in the debugger and profiler; OOP extensions to TASM and of course the inclusion of the rather wonderful Turbo C++ for Windows. Borland's compilers may not be perfect, but the release of these products means that, when it comes to C++, Microsoft has got a lot of catching up to do.

.EXE

Pricing of the new products is as follows: Borland C++ 3.0 - £299.95; BC++3.0 & Application Frameworks - £439.95; Turbo C++ for Windows - £119.95.

Existing users of BC++2.0 & Application Frameworks can upgrade to v3.0 for free. Contact Borland for other upgrade details. Borland C++ 3.0 and Turbo C++ for Windows should be in the shops for Christmas. Borland is on 0734 321150.

Compiler	run time (secs) -opt	run time (secs) +opt	.EXE size (bytes) -opt	.EXE size (bytes) +opt	compile & link time (secs) -opt	compile & link time (secs) +opt
Borland C++ 2.0	0.44	0.44	3850	3866	4.66	4.62
Borland C++ 3.0	0.49	0.49	4108	4092	9.40	10.98
Microsoft C 6.0	21.44	21.42	2923	2923	6.15	6.26
TopSpeed C 3.02	0.44	0.44	1866	1866	14.34	14.56
Zortech C++ 3.0	**0.38**	**0.32**	**1844**	**1828**	**3.02**	**3.68**

NOTE: +opt data for speed-optimised code only. Faster times and smaller sizes are in bold face.

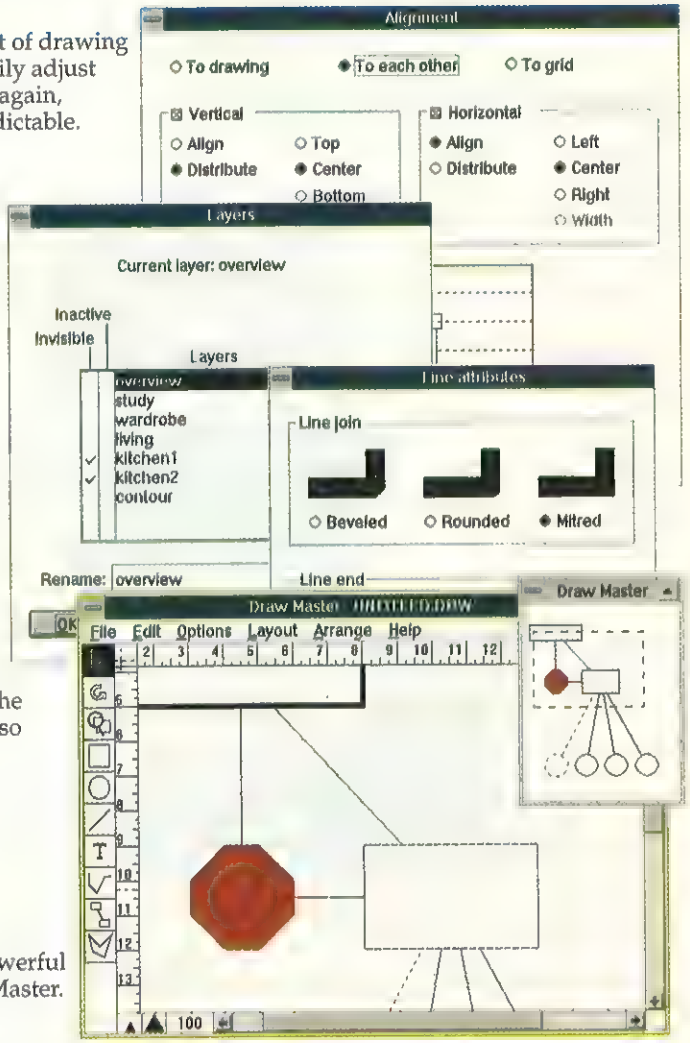
Figure 15 - NEWALLOC program results

Environment	Compile & link time (min:sec)	.EXE size (bytes) -browse info	.EXE size (bytes) +browse info
Turbo C++ for Windows	4:35	324789	331687
BC++ 3.0 (IDE in DOS box)	5:05	324789	331687
BC++ 3.0 (IDE in DOS)	4:37	324789	331687

Figure 16 - CHECKERS test results

Will DRAW_Master save you a lot of time and money too?

DRAW_Master is a construction kit for the easy development of drawing tools. If you need to build a drawing application you can easily adjust DRAW_Master to your exact needs and use it over and over again, making the development of drawing tools easy, fast and predictable.



Product of the _Master Series

DRAW_Master was built as an extension to our GUI_Master object-oriented development environment for OS/2 PM and Microsoft Windows. GUI_Master itself consists of an Interface Builder, a Class Browser and a class tree with 85 user interface classes.

High Functionality

DRAW_Master consists of a class tree of more than 120 classes. Together they form a drawing application which offers a lot of standard functionality such as undo/redo, copy/paste, zoom in/zoom out, alignment, patterns, drawing size. In addition it has features like layer control, overview window and resizing. You can use DRAW_Master as the starting point for your application. By subclassing you can adjust and expand your new application. With DRAW_Master you can build a new drawing tool in a few weeks.

Source code included

To give you the best benefit of DRAW_Master we included the source code. A cookbook and a class reference manual are also included.

Prerequisites

- C++ 2.x compiler
- SDK (for OS/2or MS Windows)
- GUI_Master (source license advised).

Free drawing tool included

If you order now you will receive our OS2DRAW tool. A powerful drawing tool in the OS/2 environment made with DRAW_Master.

Prices

DRAW_Master source code license	$15,000.=
GUI_Master source code license	$50,000.=

There are no runtime licenses

Request for Information

VSR Holland
P.O. Box 2584
3500 GN Utrecht
The Netherlands

Fax: Intl. +31 30 31 04 26
Phone: Intl. +31 30 32 49 44

Name _____

Company name _____

Department _____

Street _____

City _____ State _____ Zip _____

Country _____ DRAW_A4_0991_C

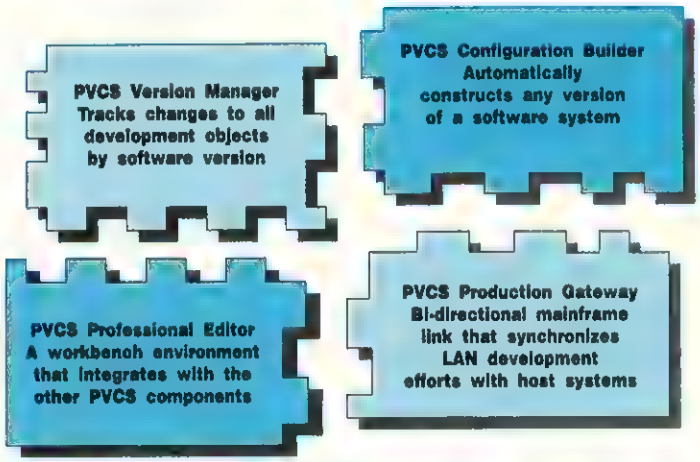
FEATURES

- Patterns
- Object selection
- Printing
- Multiple undo/ redo
- Changeable palette
- Rotation
- Zoom in/ Zoom out
- Polygon
- Connections
- Scrolling
- Auto scrolling
- Rulers
- Grid
- Reshaping
- Fonts
- Background/ foreground
- Grouping/ ungrouping
- Automatic Page numbering
- Page breaks
- Polylines
- Layering
- Colouring
- Keyboard shortcuts
- Resizing
- Copy/paste
- Duplication
- Dashed Lines
- Line width
- Line joins
- Line ends
- Grid snapping
- Overview
- Alignment
- 135 changeable classes
- Reuse of code
- Adding new classes
- Class reference
- Cookbook

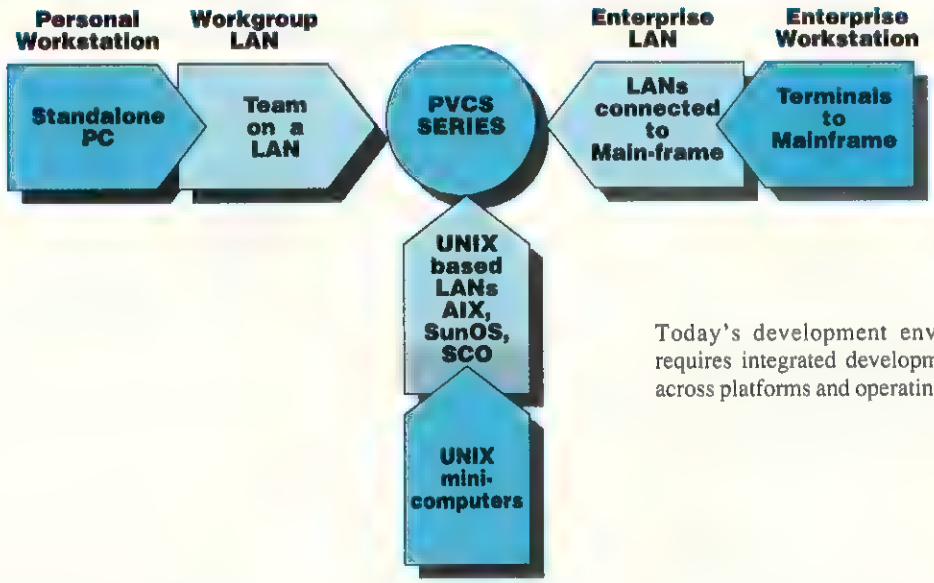
Maximise Productivity, Minimise Mistakes

To produce and maintain reliable software requires a comprehensive configuration management strategy. This strategy must support work group development on LANs, with distribution across multiple servers, wide area networks, and host systems.

The PVCS Series for Configuration Management



The PVCS Series for Distributed Environments



Today's development environment requires integrated development effort across platforms and operating systems.

Support for UNIX and heterogeneous workstation and LAN environments.

To optimise the use of resources, it is imperative for the configuration management system to distribute the required functionality to appropriate platforms. This means the manager must operate seamlessly across PC-DOS, OS/2, LANs and a variety of UNIX systems. This protects your organisation's investment in existing technology and assures compatibility for future growth.

Current prices for single users:

DOS or OS/2 or UNIX	: PVCS £345	Configuration Builder £149	PVCS Professional Editor* £99
DOS and OS/2	: PVCS £445	Configuration Builder £199	PVCS Professional Editor* £125

* On Special Offer

Tel (+44) 071 625 5255
Fax (+44) 071 624 9404

Readmar Systems
L I M I T E D

Sales and Support for Development Software

The PVCS Series by INTERSOLV

CIRCLE NO. 355

.EXE Readership Survey

About three months ago we performed a readership survey. Cliff Saran looks at some of its findings.

Congratulations if you were one of the lucky readers to participate in our survey! And many thanks if you took the trouble to answer the questions. In September this year, we took a random sample of 1000 .EXE readers and sent them a disk-based questionnaire program asking, among other things, the respondents' occupation, their qualifications and their opinions of .EXE magazine. There were also questions on the software/hardware that they or their organisation used including changes to software/hardware that would be made within the next six months. We had a dual purpose: to take a 'snapshot' view of the state of the industry, and to gather opinions and views of .EXE to help us improve it in 1992.

In this report, I have compiled a summary of a few of the more interesting revelations. But, before going into specific detail, let's start with the basics...

What sort of a person reads .EXE?

Let's construct the 'average' .EXE reader. You're probably a C programmer with between 6-10 years programming experience. You have been educated to graduate level, but you are not a member of the BCS. You have a say in the purchase of software and hardware. There's a 16Mhz 80386DX PC on your desk with 4 MB or more of memory. (My! What a lot of memory you have. We presume that you insist upon running Windows in enhanced mode). If you are connected to a LAN - and just over half of you are - then it's probably based on Novell. Needless to say, your operating system is MS-DOS.

I suppose that no-one has ever been sacked for buying Microsoft either

With the PC manufacturers offering ever faster, more powerful machines, I was surprised that so many developers were using comparatively sluggish machines. Perhaps everybody bought new 16MHz 386s when that was 'state of the art', costing more than an arm and a leg and now they can't afford to upgrade to something better - especially as an old 16MHz 80386DX can run Windows properly and can be used to develop all PC-based software, including 32-Bit applications (sigh of relief from the accounts department).

You are quite loyal to .EXE - although you might take *Byte* or *PCW*, you probably don't read any other specialist programming magazine, whether it be US or UK published. Thanks. You're quite *old*, if you don't mind me saying so - between 35 and 44 years. There goes the myth of the industry built on brat geniuses. And, oh yes, you're a bloke. She-programmers account for less than 2.5% of the population. Boo-hoo.

What is your main Operating System?

No surprises here. Over 80% of respondents chose MS-DOS (see Figure 1). There were significantly more UNIX users compared to OS/2 users, notwithstanding the alleged popularity of OS/2 among developers for running PWB. OS/2 weenies, before writing more fierce letters (see .EXE November '91), should take a peek at the next section.

Would you adopt a new Operating System?

Quite a few of you will be on the move within the next six months. Twenty percent of the respondents indicated that they

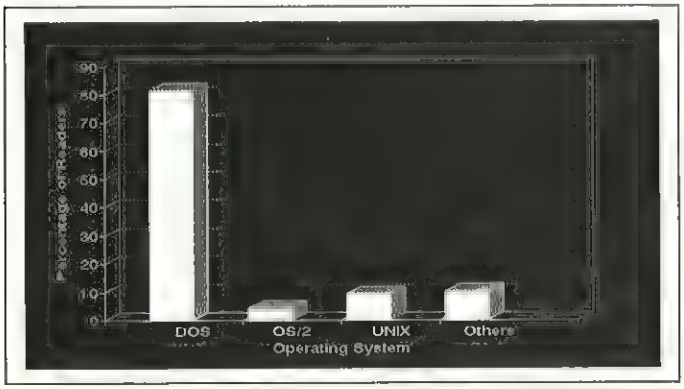


Figure 1 - Main Operating System

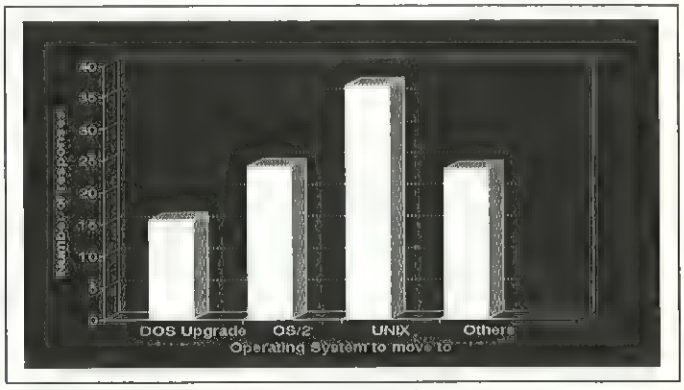


Figure 2 - The move from DOS

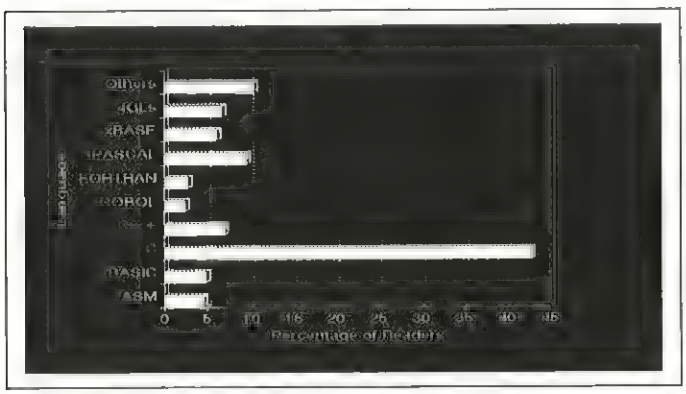


Figure 3 - Primary Development Language

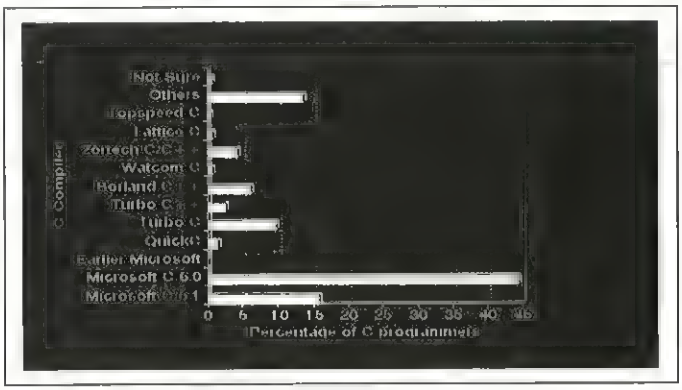


Figure 4 - Main C compilers used

would be adopting a new operating system. In Figure 2, as expected, more respondents said that they are considering a move to UNIX than to OS/2 - however, it is a proportionally strong showing by Big Blue's baby (presumably highly influenced by the forthcoming V2.0 which, at the time the questionnaires were answered, was due to appear before the end of 1991). There were also a few readers who were (superficially mysteriously) contemplating a change from DOS to DOS (presumably they intended upgrading to MS-DOS 5.0 or DR-DOS V6.0 or some such).

What is your main development language?

Yip, it's C. As Figure 3 clearly shows, C was, by far, the most popular language, with more than 40% of respondents choosing it. The results for C++ indicate that take-up of that language is still at an early stage - it has yet to overtake even Pascal. A possible explanation for this is perhaps implied by the analysis in Figure 4. A follow-up question, 'Are you satisfied with your main development language?', drew a resounding 93% 'Yes', which suggests the software developers suffer from heavy brand loyalty - which again has implications for the results of Figure 4. The proportion of xBASE users shown in Figure 3 is unrepresentative. Many respondents failed to appreciate that xBASE included Clipper, or FoxPro, or whatever, and answered 'OTHER' instead. We have not yet had time to go through all the answers and fix this, but I estimate that about half of the 'OTHERS' belong to xBASE.

Out of 131 responses :-

	Good	Bad
Microsoft	44	47
Borland	16	1
Zortech	5	0
Others	13	5

Figure 5 - Table of C compiler Technical Support Ratings

How it was done

'Magazines ... are notorious for announcing biased results based on responses to questionnaires appearing in the periodical. These informal polls rarely come with confidence intervals or any details of the methods used, so the problem of self-selected samples is not always immediately apparent.' from *Innumeracy* by John Allen Paulos, published by Penguin Books (1988).

Surveys are notoriously difficult to get right. In a famous case in the US, a group of people was asked the following question: 'Did you, as a good citizen, vote in the recent county elections.' Over half of them said they had. The catch, as you will already have guessed, was that no elections had been held. Nobody wanted to be a Bad Citizen.

It's easy to bias the results of a questionnaire accidentally by asking a weighted question. But even before you even get that far, you must address the question: 'Who do you ask?' The typical approach to readership surveys is to include a paper questionnaire in the magazine. The problem with this is that the people who bother to answer (always a very low percentage of the whole population) are *keen* - they have some ulterior motivation for replying. Offering some bribe to respondents, eg holding a prize draw for a colour TV as has been done in previous .EXE surveys - improves the number of replies, but distorts the sample in other, more subtle, ways.

This is why Carne Martin, the market research company employed by .EXE to carry out the readership survey, devised a different approach. One thousand people (about 1/17th of the current readership) were randomly chosen from the .EXE subscriber database. To each of these was mailed two MS-DOS disks (3.5" and 5.25"). A Turbo Pascal program on these disks (written by the Editor exploiting persistent objects to hold the 180-odd screens) acted as an interactive questionnaire form.

There turned out to be many advantages to the computer-based approach. The questions were structured so that, for example, someone who used Pascal as a primary development system wasn't then pestered with questions about C compilers. This permitted a depth of detail not normally possible. On return of the disk, the data could be sucked into a database without costly and error-inducing re-keying. The anticipated disadvantages - such as company anti-virus security forbidding the use of unsolicited software, or Watts' software falling over - turned out not to be significant.

After only four weeks, over 40% of the 1000 disks had been returned. A reminder was then mailed and, after a further month, there was a total of over 650 responses in the master database. This is quite a considerable feat. With multiple reminders, the response from a typical survey is in the order of 35%-45%. The .EXE readership survey achieved over 60% using only a single reminder (see Figure 8).

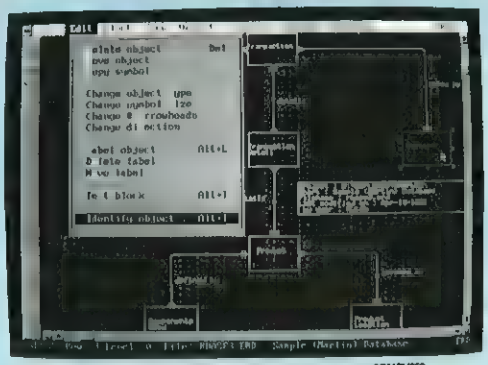
We are proud of our survey, and feel that the results that we have obtained are an unusually reliable indicator of the state of our industry. Thanks once again to those 650 who were generous enough to give us a little of their time.

EasyCASE™ Plus

NEW! VERSION 3.0 Call for upgrade details.

The affordable approach to software engineering... only $495*

Finally, there's a CASE tool that won't get in the way of your creativity... A tool that makes structured analysis, structured design and data modeling as easy as working with any other tool on your PC - EasyCASE Plus! Using EasyCASE Plus' new, easy to use graphical user interface (GUI), you'll be creating and editing charts, linking them, and building your data dictionary in no time. As well as being easy to use and easy to learn, EasyCASE Plus is easy on your budget! Ask any user. They'll tell you it's the best buy for your PC based CASE tool needs. Discover why over 4,000 software professionals use EasyCASE Plus and how you can join them!



"EasyCASE Plus is a well designed, low priced tool that is easy to learn and provides excellent diagramming capabilities... EasyCASE Plus is an excellent investment." COMPUTER LANGUAGE PRODUCTIVITY AWARD 1990

Features:

- IBM SAA/CUA compliant graphical user interface (GUI)
- Extensive diagram editing features
- Integrated dBASE III compatible data dictionary
- Integrated dictionary manager, reports manager, process editor
- Hierarchical chart linking & process decomposition
- Record and element definitions
- Extensive printer, plotter and desktop publishing support
- Data dictionary import, export, and merge
- On-line help
- Comprehensive documentation with tutorial
- Access to your database, word processor, DOS, etc.
- Integrated diagram analysis (optional)

Requirements:

Runs on: IBM PC or PS/2 (AT recommended), DOS 3.1 or higher, EGA/VGA color, mouse, 640 K RAM (500 K free), 1 MB EMS recommended, math co-processor supported. Printers/Plotters Supported: Epson FX & LQ, IBM Graphics & Proprinter X24, HP QuietJet, DeskJet, & LaserJet, HP Plotters, PostScript.

EasyCASE Professional $649*
(includes integrated DFD level balancing and data dictionary/diagram analysis)

Methods:

- Yourdon/DeMarco
- Gane & Sarson
- SSADM (DFDs)
- Ward-Mellor/Hatley
- Yourdon/Constantine
- Martin
- Chen, Bachman

Diagram Types:

- Data Flow Diagrams (DFDs)
- Structure Charts
- State Transition Diagrams
- Entity Relationship (ERDs)
- Data Model Diagrams
- Transformation Schema (real-time DFDs)

Evergreen CASE Tools
16650 NE 79th Street
Suite 200
Redmond, WA 98052
USA
FAX: (206) 883-7676

MasterCard VISA

Call today for a brochure!
Tel: (206) 881-5149

CIRCLE NO. 356

Desktop FORTRAN

EXPERTS AGREE WHAT TO BUY – THEY JUST CAN'T AGREE WHY!

"The . . . FTN77 compiler has allowed us to port our main-frame applications to 386 micros with minimum effort. The speed of compilation is incredible!. . . The run time performance is fantastic, full support for 32-bit operations and the 80387 coprocessor means our programs run like lightning!"
Tony Fitzpatrick, Exploration Consultants Ltd

"I find the FTN77/386 compiler to be fast and easy to use . . . . We often download programs under development from a mainframe to the PC to run through this computer."
Rob McLaren, FTN77/386 User.

"Without the FTN77/386 compiler it would have been totally impractical for us to convert our million-line + power system analysis program (PSS/E) to run on a 386 PC"
Wayne B'Rells, Power Technologies Inc

"FTN77/386 provides more intrinsic functions than any other functions to manipulate strings, handle screen and keyboard I/O, read the date and time, parse the command line . . . Advanced file manipulation functions allow an FTN77/386 program to work with files in a C-like manner".
Scott Ladd, Computer Language, Nov 1988

". . . I am very impressed and have decided to shift all our production programs over to FTN77 . . ."
Dr. G.R. Chaplan, NCT Investments Ltd

"The ease of use, speed, debugging facilities, dynamic libraries with resulting small .EXE files, and extensive library routines made it a joy to use."
H. Gordon Jensen, FTN77/386 User.

"By far the fastest and most complete FORTRAN compiler that I have ever seen or used in the DOS environment".
Tim Wool, FTN77/386 User

"The debugger was very easy to use. There is an on-line help facility which made reference to the manual unnecessary."
. . . Mike Gunn and Arid Brutto, EXE Magazine, May 1989.

"For sheer speed and sleekness the (Salford) compiler stands out like a Ferrari among Yugos."
Jack W. Crenshaw, Computer Language, May 1990

FAST, FRIENDLY & FULLY FEATURED – *THE PERFECT MIX – CALL US NOW!*

SALFORD SOFTWARE LTD • VENABLES BUILDINGS • 5 COCKCROFT ROAD • SALFORD M5 4NT
TEL: 061 745 5678 • FAX: 061 745 5666

FULL WINDOWS 3.0 SUPPORT!

FTN77 386/486 – extending MS-DOS.

UNIX VERSION AVAILABLE TOO!

CIRCLE NO. 357

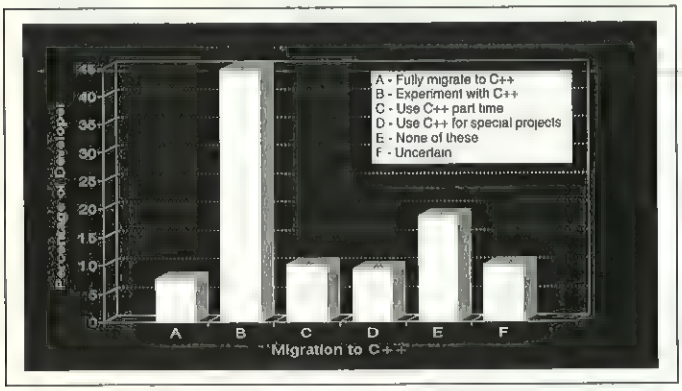


Figure 6 - Migration to C++

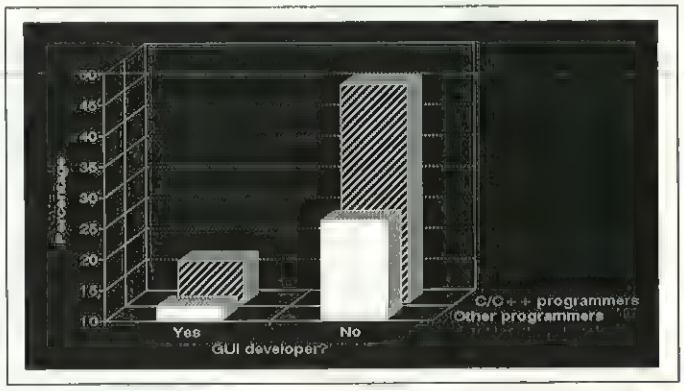


Figure 7 - Percentage of programmers developing for GUIs

Which C compiler do you most often use?

Microsoft is the outright winner (see Figure 4). Although C V6.0 was hailed in some quarters as a rather disappointing successor to V5.1, it easily secured a colossal 44% of the C-programmer response. (Note, however, that a third of Microsoft users have failed to upgrade.) With Zortech and Watcom both producing compilers with the ability to generate 32-bit code, JPI offering the elegant TopSpeed system and Borland offering new releases every ten days, Microsoft's supremacy remains undented. Remember what they used to say about IBM? I suppose that no one has ever been sacked for buying Microsoft either.

How do you rate technical support on your C compiler?

Actually, we asked the users of all languages about technical support; but only in the case of C did we obtain sufficient numbers to obtain any meaningful results. Microsoft's supremacy in the market, however, was not reflected in its tech support. The table in Figure 5 indicates that .EXE readers who have been writing us irritable letters were probably not isolated cases - about one half of the callers to Microsoft were dissatisfied. Unfortunately, the response rate for the other manufacturers was not high enough for us to be able to say whether they did any better, so it is perhaps a little naughty of us to publish their figures. We wonder if Microsoft's performance has anything to with its computerised telephone answering system which has certainly caused frustration in this office. We hear that Borland UK is to install a similar system...

Magazines ... are notorious for announcing biased results based on responses to questionnaires

What are your plans for migration to C++?

This question was just for those who used C as their primary development language. Given that very few of the respondents actually used C++ as their main language, it is not surprising that only 7% were committed to moving over to C++ (see Figure 6). However, many more were intending to dabble with C++. Of the 280 replies, 179 respondents said that they would be using C++ within the next six months (ie over 64%).

What are your plans for developing GUI applications?

We've all got Windows applications running on our machines, but there seems to be very few developers writing GUI software (perhaps you've heard that it is quite difficult). Just over 20% of those asked, said they developed GUIs in C/C++ (Figure 7). Less than 10% used other languages for GUI development. In retrospect, 45% of software developers in the survey were C/C++ programmers writing non-GUI applications. Given that it is such a tortuous approach, it is remarkable that most GUI developers probably write their applications in C (139 out of 309 replies). For those opting for C++, 50 said they would use a 3rd party GUI class library while 45 said they would develop an in-house library (still working on the wheel, I suppose). Why? Not good news for manufacturers of libraries such as CommonView, C++/Views, ObjectWindows etc.

Conclusion

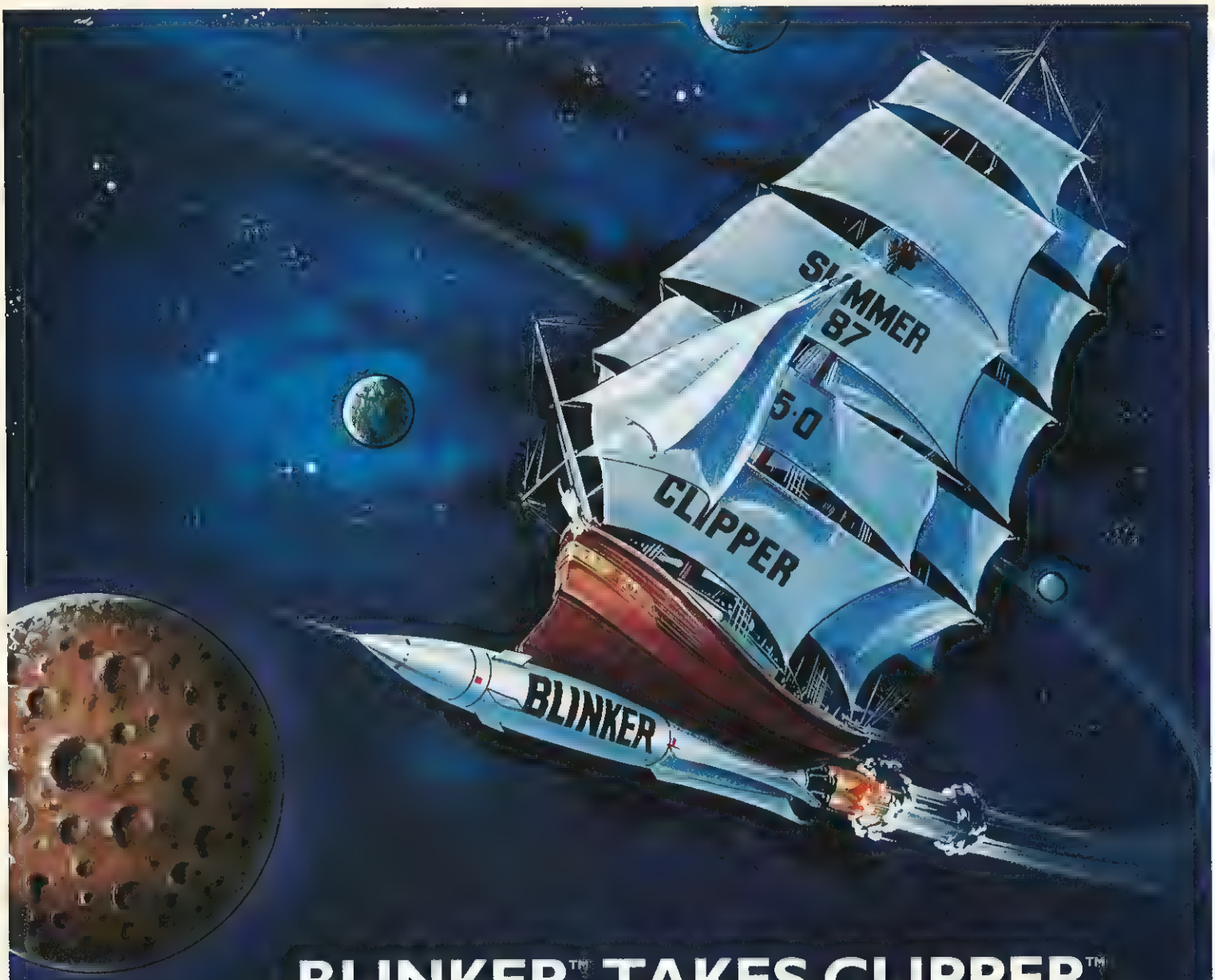
This is the first time that we have attempted a survey of such thoroughness. We believe it to be unrivalled in the field. Of course, the real fruits of our efforts will begin to appear the *second* time we perform the survey, when we will be able to monitor trends. Meanwhile, the current data set confirms a few prejudices, upsets a few applecarts, but above all gives us a better idea of what you need from .EXE magazine. See you next year.

EXE

Both the data collected in this survey and the software used are available to interested third parties. Please contact Jon Howell on 081 994 6477 for details.

1991	.EXE Reader Survey (Carne Martin)	67%
1991	BUSINESS Reader Survey (NOP)	47%
1990	European Businessman Readership Survey (RSL)	48%
1989	International Finance Managers in Europe (RSL)	46%
1987	Building Centre Readership Survey (CCMI)	43%
1989	Banner Computer Readership Survey (Codex)	42%

Figure 8 - Response rates from other surveys



BLINKER™ TAKES CLIPPER™ OUT OF THIS WORLD

Blinker eliminates the need for complex overlay structures for both Clipper Summer '87 and 5.0

* Significantly reduces run-time memory requirements
* Accelerates your development cycle 3-10 times faster than any other current linker and even faster when linking incrementaly
* Dynamically overlays C and Assembler code sich as Extend.lib, dGE, FUNCky, other Clipper Add-ons and your own 'C' code
* Optimises your application with advanced profiling functions
* Includes memory defragmentation, 'burning-in' of Clipper environmental variables/serial numbers and creation of demonstration versions

IT'S ALL PLAIN SAILING WITH BLINKER

What the experts say

Here are just some comments from the 5000 happy users to date

"It's blinking fast!"

"Congratulations – we gained over 100k"

"It's so good, I went through all my programs and re-compiled/linked just for the pleasure of seeing all the free memory"

"It's fast! Handles overlays like a dream"

"Amazing – No more messing about with overlay structures!"

"Brilliant!... But my coffee breaks have been shortened!"

"Actually exceeded our most optimistic performance hopes! (just for once)"

European sales and support:

QBS Software Limited

10 Barley Mow Passage, Chiswick, London W4 4PH
Telephone 081-994 4842. Fax 081-994 3441. BBS 081-747 1979

Authors and publishers:

Blink Inc Limited

PO Box 308, Cheltenham
Gloucestershire GL50 4PL

Software sans Frontières

National language support is usually added to software as an afterthought. BJ Thomson highlights some of the problems you will encounter when you try to do it.

In the April 1991 issue of .EXE, Ebbe Snderby gave us an insight into the problems suffered by our Scandinavian colleagues when computers became widespread. I have been researching the subject of codepages/character sets for a customer, and the following may be of interest to those producing software for the international market.

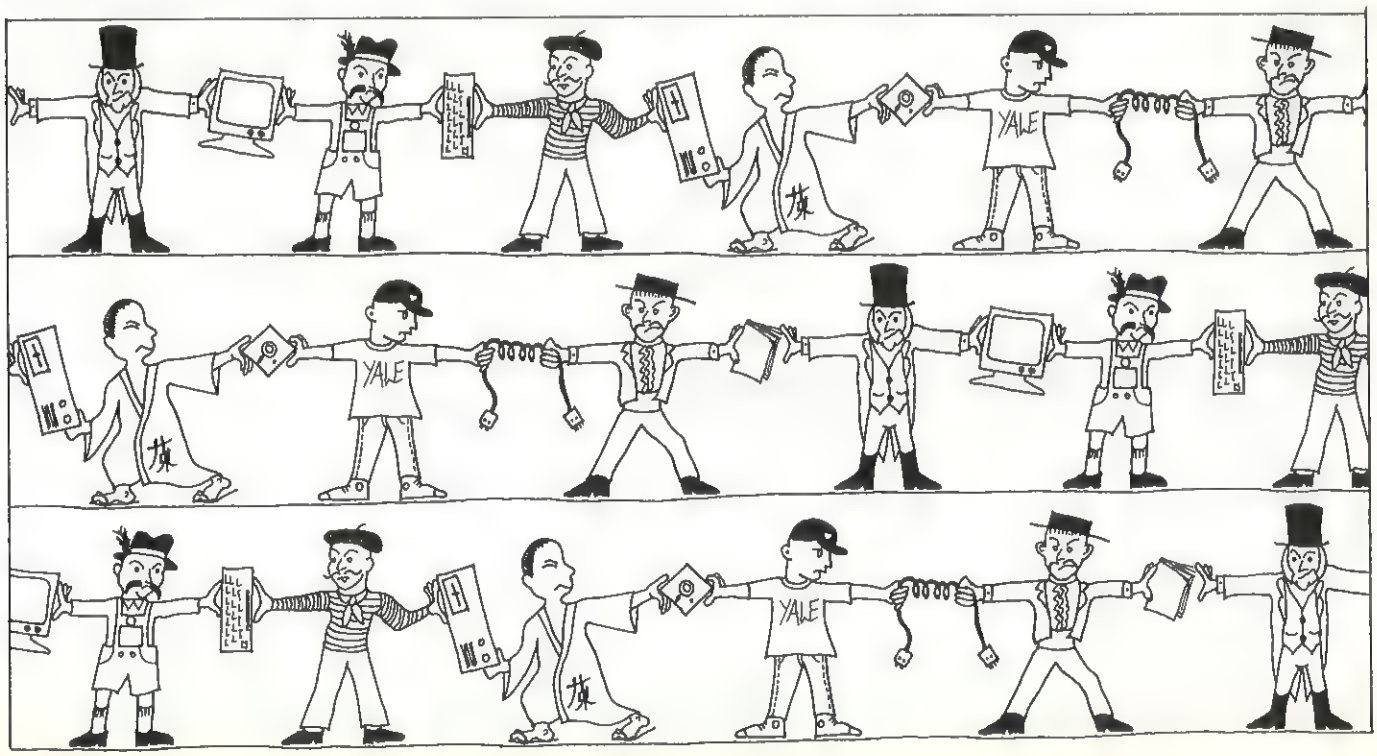
History

First, a little history. When IBM produced the first PC, it was never expected to become the juggernaut that it has. The design team produced a machine for what was considered to be a tiny market. It was of course American and, with typical American parochialism, its display and printer character sets only included characters in common use in America. This character set, now blown into the ROMs of 150-200 million PC video adapters, printers and plotters, became known as 'codepage 437'. As the PC became the standard machine for use in business, it eventually escaped onto the British and European markets, and the European support organisations quickly encountered the kind of problems related by Mr. Snderby.

I trust you will now be well-equipped to plunge into the m☆l♥e of writing multilingual software

It was in fact the Scandinavian countries who started the PC codepage saga, when the local IBM division registered codepage 865. A codepage is simply a collection of 255 characters, the first 128 normally being fixed, and known as the ASCII set. A codepoint is simply the position within the codepage. The high 128 characters are variable from one codepage to another, so for instance, the Sterling (£) symbol is codepoint 156 decimal in codepage 437, but if I send a decimal 156 to my Japanese Epson clone, it prints a blank. One can imagine that this caused a little consternation when banks started sending electronic mail to each other across national boundaries.

To resolve the above problems, IBM introduced codepage 850, in which the range 128 to 255 contains the majority of accented characters used worldwide. There is a limit



to what you can do with 128 characters of course, but more on that later. Codepage 850 is the recommended codepage for all new installations. Unfortunately, to maintain compatibility with existing kit, MS-DOS defaults to 437 and 850 has to be explicitly loaded when your friendly dealer installs your shiny new PC. This is a fine example of the gap between idealised corporate policy and real life. Since the many PC dealers find editing CONFIG.SYS rather a strain, 850 doesn't get installed, and the saga continues.

Nowadays, most printers at least support 437; some, like those supplied by IBM, support downloadable codepages, and it is not normally too difficult to get the printer to print what you see on the screen.

ANSI has a go

There is a simple answer, of course: just get everyone to agree on a single character set for the future. The American National Standard Institution tried to do just that, and defined the codepoints 128 to 255 to contain most of the characters found in codepage 850, and a few more. It didn't want to offend DEC, so it didn't standardise on IBM codepage 850, but went its own way. This is the character set now used in Windows.

Now Windows, as you know, currently runs on DOS, and so our European friends have a very interesting situation. They can produce a file in, say, Wordstar, containing characters like 'Ç' (C cedilla) which is codepoint 128 on codepage 437. When this file is loaded into Windows Write on the same machine, it will display a block character, because the Windows character set does not support codepoint 128.

The American National Standard Institution didn't want to offend DEC, so it didn't standardise on IBM codepage 850

Windows confusion

But what about just Windows by itself? Provided you stay in Windows, all your problems go away. You install Windows once, for your hardware, and What You See Is What You Get. All Windows printer drivers understand the Windows character set, and if you change your printer, you simply install a new driver, and all Windows applications will print correctly.

The same applies to screen drivers. In order to place accented characters in Windows applications, you may use the old Alt-Keypad approach - just hold the Alt key down, type 135 on the keypad, and release the Alt key. You should now see c cedilla (ç). But... looking at page 568 in the Windows 3.0 user manual, you will see that in the Windows character set, codepoint 135 is a block character... codepoint 135 is c cedilla in codepage 437 on the facing page... ie Windows is translating your keypad input to an entirely different value. Gulp.

Time to start poking around the grubbier functions buried in the Windows API. You may have seen a couple of functions called `OemToAnsi()` and `AnsiToOem()`. These functions are there for us to use in translating from codepage 437 to Windows (ANSI) and vice versa. These functions are used internally for translating filenames, so that we don't save files that we cannot load outside Windows... So what happens if, like a good boy, you re-installed DOS properly after your dealer left, and your PC boots in codepage 850, you then install Windows, and save a file with a name containing accented characters, and display the directory under DOS? Yup, you guessed. See Figures 1 and 2.

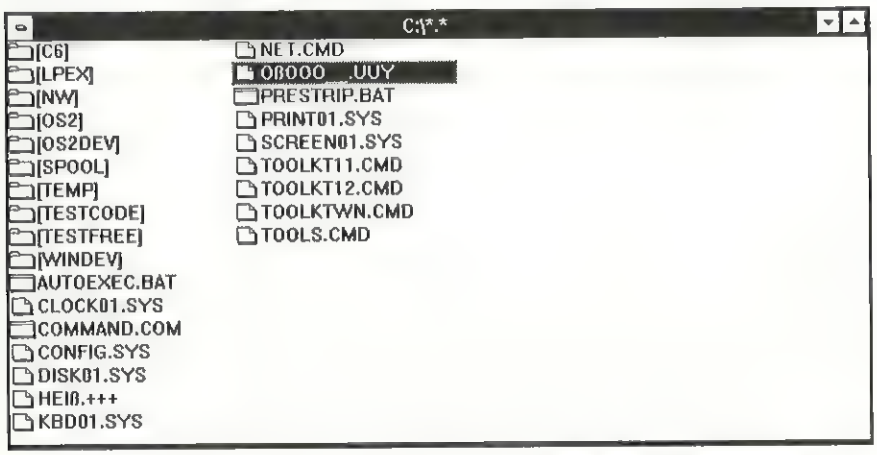


Figure 1 - There's a strange character in the directory listing...

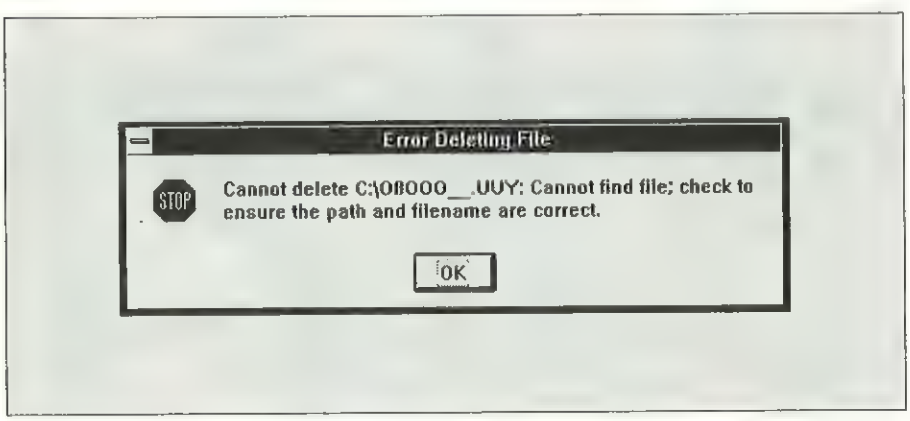


Figure 2 - ... and it's impossible to get rid of it

OS/2 fix

Never mind, it's all fixed under OS/2. The default installation for OS/2 sets up codepage 850, and the PM character set is 850 compatible. Now what happens, if, like me, you develop for Windows under OS/2 (because I can edit the source with LPEX, then start the compiler in a windowed command prompt, and it doesn't stop when I change focus, and it doesn't run out of far heap space). I can then run Windows in the DOS box, and accented characters... AGHHH!

Now what are they going to do when we get Windows applications running on OS/2 via the Binary Compatibility Layer? or even native under OS/2 3.0? AGHHHH[2]! Notwith-

Software errors can be brutal...
...find them first with Evaluator

Testing Software properly means hours of boredom, and is very, very expensive. But letting users find errors can be as bruising as an encounter with our friend on the left.

Evaluator can help. A unique testing tool, Evaluator can automate the entire testing process, including regression, stress and acceptance testing. It can do this unattended, leaving staff free to perform more productive work.

Evaluator will test every kind of software including GUI applications such as Windows 3. Operating system independent you can test DOS, OS/2 or UNIX applications. If you can get your mainframe application to display on a PC it can test that too!

Evaluator is non-intrusive, which means that the machine you test your software on will be the same as the machine it gets used on; a very important quality consideration.

So if you don't want any unpleasant surprises from your software, call The Products Group for full details on 0285 - 655888

ELVEREX

QA

Tick here for full details:

☐ Evaluator ☐ Consultancy
☐ Product Catalogue ☐ Training: OS/2, UNIX, Windows, C, Object Oriented, Networks.

NAME _____ TITLE _____

COMPANY _____

ADDRESS _____

TEL. NO. _____

EXE 12/91

Mail to: The Products Group, QA Training Ltd., Cecily Hill Castle, Cirencester, Gloucestershire GL7 2EF. UK.
Tel: 0285 655888 Fax: 0285 640181

CIRCLE NO. 362

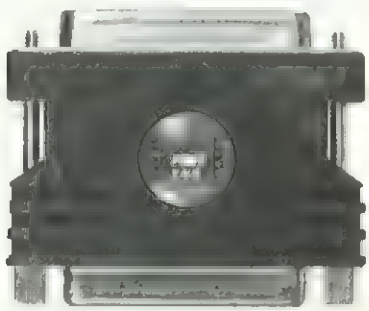
5 out of 5 hackers prefer other software protection methods to Hardlock E-Y-E®

What hackers dislike...
Hardlock E-Y-E was designed using cryptographic principles. It took the experience and know-how of Germany's No. 1 in software protection and the leading edge technology of a US semiconductor company to create the ultimate software protection tool. Hardlock E-Y-E is based on a custom chip featuring secure algorithmic response rather than simple bit swapping or counting schemes.

What software developers like...
Hardlock E-Y-E combines all the features software developers require in a single product: algorithmic response to provide security and an optional non-volatile memory to allow custom configurations. FAST Electronic has made implementation of Hardlock E-Y-E in your software easy. Use HL-Crypt to protect .EXE or .COM files, or incorporate high level language interface routines in your software. The algorithm parameters and the contents of the memory can be programmed in seconds using our Crypto-Programmer card. This unique card guarantees that no one else can burn your original codes. Simply plug the card into any PC slot and start up your own Hardlock E-Y-E workshop.

What your customers will like...
Hardlock E-Y-E allows unlimited backup copies. The device is shipped with the software for the user simply to plug into the parallel interface and forget. Daisy chainability, outstanding reliability (no battery is needed), and the most compact High-Tech design ensure that your customer will accept Hardlock E-Y-E.

What your accountant will like...
Hardlock E-Y-E needs no factory coding. This ensures optimum delivery schedules and stock flexibility. Revenues will go up as software piracy and multiple usage are prevented. Despite its wealth of features, Hardlock E-Y-E's prices remain competitive.



Hardlock E-Y-E programmable, algorithmic response and memory option – all in one.

...As more and more software developers, customers and accountants appreciate the Hardlock E-Y-E device, hackers like it less and less.

MAGNIFEYE

Order your demo unit today. Contact Magnifeye,
235-239 Walmer Road, Walmer Studio #6, W11 4EY, Telephone 071 221 8024, Fax 071 792 3449.

CIRCLE NO. 363

standing the above, I believe OS/2 and Windows, with some tweaking, are the way to go.

IBM Orient

I recently had to modify a software product to work on Asian hardware. IBM have for some time produced machines aimed at the Asian market. I had a model PS/55 (no, this is not a misprint) to play with, and was surprised just how seriously the company takes the business of producing International software. A PS/55 is a desktop machine, with a 386/20 processor, acres of RAM and a 1024/768 display with an intelligent graphics adapter. The keyboard is *very* impressive, with half a dozen shift states, and more characters per key cap than a Sinclair Spectrum. The machine boots under codepage 932, a combined codepage.

As you probably know, the Asian languages have many thousands of characters in daily use, and so we cannot simply define a new 256 byte codepage. IBM has two methods of supporting Asian characters. The first, used on mainframes, uses a Shift-In, Shift out approach. The terminal normally works in a single byte EBCDIC codepage, but when it sees a Shift-in character it assumes that all characters following are Double Byte until the next Shift-Out character.

The second method, used on OS/2 (and maybe Windows? check out `AnsiNext()` and `AnsiPrev()` - don't bother asking tech support) is rather more interesting. Codepage 932 is 256 characters in size. The first 128 are as usual, the same as ASCII - more or less. Some of the high bit characters are reserved for use as the first byte of a Double Byte Character - a DBCS1. There is no restriction on the second byte - the DBCS2, save that it cannot be NUL `'\0'`. There is an OS/2 function, `DosGetDBCSEv()`, which returns an array of bytes describing the start and end points of these DBCS1 ranges. Armed with this information we can scan strings from the first character, and if the byte is within one of these reserved ranges, we know that it is the first byte of a DBCS character. If so, then to reach the next character, we step two bytes instead of one.

There are various restrictions on what we may do with DBCS characters. For instance, when marking a block, we must always include both bytes, we must never wrap at a point which would split a DBCS character, and if the DBCS2 falls outside our display area, the DBCS1 must be blanked. In Japanese, there are also KINSOKU characters, some of which must not be placed at the start of a new line, and others which must not be placed at the end of a line. Some Asian languages also place restrictions on insertion of space characters in strings.

When this file is loaded into Windows Write on the same machine, it will display a block character

UNICODE

In the future, we may see the new UNICODE standard being implemented. Not surprisingly, this has the whole-hearted support of IBM and most other hardware manufacturers, because it instantly doubles the storage requirements for text - characters are 16 bits wide instead of 8. The rest of us may balk at having to replace our hardware again. Then there is the UNIX camp to consider. Apparently, the next version of UNIX, whatever that is, will offer two methods of supporting multi-byte characters - the first reserves a character with the MSB set as a DBCS1, and thus, by inference the following character is a DBCS2. The second method reserves some characters as Shift indicators, like EBCDIC. But of course, most UNIX boxes don't use EBCDIC, so it won't be the same as the IBM method. (AGHHHHHH^3!)

Tips

Finally, here are some suggestions on making more easily portable code.

1) If you must use strings, place them in a separate file, eg in Windows and PM, use the `STRINGTABLE` statement in a resource file.

2) When dealing with user entered text, avoid code which explicitly increments and decrements `char` pointers, instead provide your own functions to step to the next or back to the previous character. (Within the function, you can simply increment and decrement a `char` pointer, if you *know* that the code will never be ported to another environment). When it *is* ported to another environment, the functions can easily be replaced with working code. If you work in Windows, use the `AnsiNext()` and `AnsiPrev()` functions. For most European languages, these are very simple and fast, and if you use them, your program stands a reasonable chance of running on Japanese Windows. If you work on OS/2 PM, you will need to write your own `AnsiNext()` and `AnsiPrev()`. Hint: In your `Init()` use `DosGetDBCSEv()` to fill a global byte array[256] with `0x1` for SBCS and `0x2` for DBCS1. Then when scanning strings, simply increment the char pointer by the value in the array using the current character byte as an index. Scanning backwards is a little more complex. You must keep track of the start of the string, and scan forwards to the character before the current one. Mind your segments.

3) Note that `toupper()`, `tolower()`, `isupper()`, `islower()` etc will not give the correct results when used on accented characters, since these appear in the high 128 bytes of the codepages. In Windows, use `AnsiUpper()`, `AnsiLower()` etc.

3) In DOS, do not use the double line box characters in codepage 437, since they display real accented characters in other codepages. Most of the time, it's obvious that they are meant to be boxes, but it looks amateur. Above all, do not assume anything about the user's machine. In the installation process, ask silly questions about currency symbols etc.

4) Do not change the user's data from one codepoint to another without asking first. This is very relevant to communications software.

5) This one applies to all of us, not just programmers. When sending electronic files over wire, clarify all currency symbols with some text eg £ (Sterling) or $ (US dollars).

That's it. With this advice, I trust you will now be well-equipped to plunge into the skirmish of writing multilingual software, or as the French would put it, jump into the m☆I♥e.

EXE

BJ Thomson has been in the electronics business for about 12 or 13 years, originally as a hardware type, then more recently as a software person. With his wife he runs a small business, based in Yateley, producing technical manuals and bespoke software. He claims to be one of the few people in Britain outside IBM who has ever had to code for Double Byte Character Sets.

Sounds Peculiar

With the scent No-Needle-Drop in the air, we thought it might be fun to investigate a rather peculiar category of PC software. Will Watts has been sounding out the packages.

Before these two packages, I would have sworn - nay, laid down real money - that it wasn't possible. I run a middle-aged AST 386 which, although I believe to be wonderful, had certainly not previously shown any noise-reproduction skills. It went beep! when it booted up, and beeeeeeeep! when you left a book on the keyboard - and that was it.

So it came as a shock to discover that my AST, in common with nearly all others cast from the IBM mould, possessed the ability to reproduce recorded sound and to synthesise speech, I was most surprised.

SoftSpeak

SoftSpeak II+ is the product of Quantech Ltd, of Newcastle-Upon-Tyne. There are two components to SoftSpeak (available separately or together): the software, which lets you edit, manipulate and generally fool around with pre-recorded sound; and the hardware component, which provides a mechanism for getting sound into your PC in the first place.

Let's do the software first. SoftSpeak arrives on five 360 KBs, and contains an install program which un-LHARCs about 2.5 MB of files into the directory of your pleasure. To get the system you will need a PC running at least 8 MHz (10 MHz if it is an XT type), a hard disk (digitised sound takes up a **lot** of room - a 360 KB file of mine contains about half a minute's worth of speech), an EGA or VGA (to allow you to run the spectacular SE350 editor, illustrated in Figure 1) and a mouse (to operate the same).

The SE350 program is definitely the main item. The large horizontal window towards the top of the screen displays the waveform of the sound sample currently loaded. In effect it is a graph of the amplitude of the recorded sound (vertical access) plotted against time (those who understand their physics better please forgive my naivety). The two vertical bars, positioned about halfway across the sound buffer window, are cursors which can be moved and used to zoom in on a particular portion of the buffer. To play back the portion of the buffer between the bars, one simply clicks on the PLAY button. The sound reproduction is startlingly good on some PCs, not so fab on others - experimentation showed that it depends very much on the type and position of the speaker within the PC.

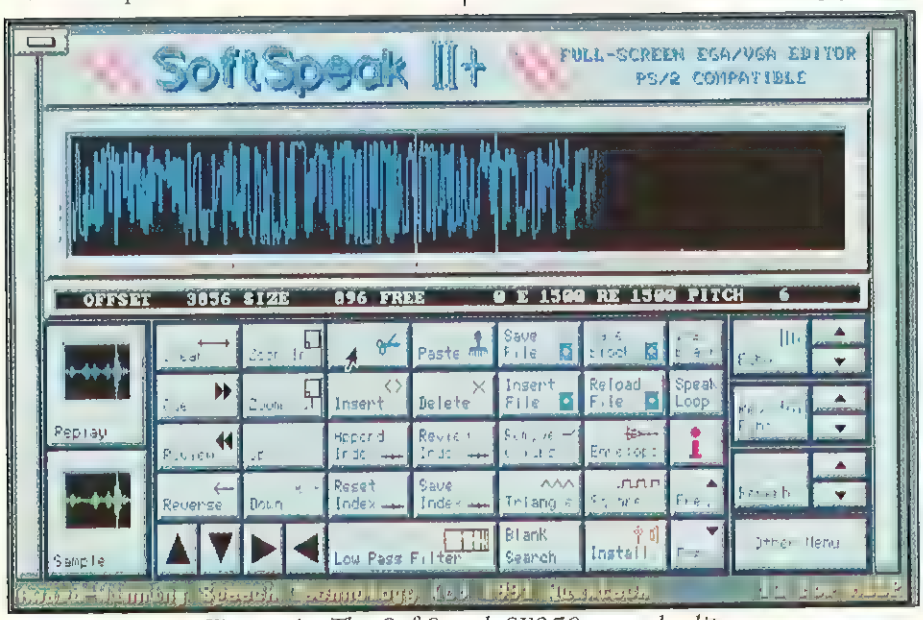


Figure 1 - The SoftSpeak SE350 sound editor

SE350 boasts a large armoury of special effects with which you can manipulate the sound. REVERB makes the difference between recording speech in a toilet cubicle and in the village hall. ECHO is the same effect, but over the extended range from toilet cubicle to carefully chosen cave. REVERSE ECHO produces an echo sound before the original, producing an effect reminiscent of the (literally) diabolical sounds from the key scenes of Hammer House of Horror films. EXPAND pumps up the volume, using non-linear amplification to cheat the limited capability of the PC's sound system. REVERSE is just the same effect as playing Sgt Pepper's backwards on your gramophone. There are about half-a-dozen other effects you can produce.

If you have the Quantech hardware, you can record directly into SE350, monitoring the recording as you go. The hardware consists of a head-set microphone, plugged into a battery driven A to D box, about the size of two packets of cigarettes, which in turn plugs into the PC's parallel port. Once you have completed your recording (the length of which is limited by the amount of conventional RAM available to SE350) you can apply any of the effects listed above to it, as well as cutting and pasting sections, removing RAM-wasting silences etc. When you are satisfied, you may save the whole lot to disk.

SoftSpeak supports two type of sound files. SOF files contain a block of sound, which must be played back as a whole. SIF files contain variable length sound 'records', these are indexed by keywords. The obvious thing to do is to associate the sound of the recording of a given word with that word. Both SIF and SOF files can be pre-

SOFTWARE

BESPOKE / MAINTENANCE / CONSULTANCY

We are a PC-Centric company Specializing in System Consultancy / Bespoke development.

LANGUAGES

- Clipper 5.01
- dBASE III + ,IV.
- MICROSOFT C
- RETRIEVE 4GL.
- BASIC.

APPLICATION AREAS

- Accounting Systems.
- Recruitment Software.
- Administrative Systems.
- Relational Databases.
- Property Management Software.

PLATFORMS

MS-DOS **NOVELL NETWARE**

London Computer Centre Ltd.
INCORPORATED 1980.

41 Grafton Way, London W1P 5LA.
TEL NO: 071-387-4455 *FAX: 071-387-6315*
CONTACT: Kevin A Smith *Shahid R Malik.*

CIRCLE NO. 364

FORTRAN

"FTN77/386 is simply the best FORTRAN compiler on *any* platform, *anywhere* - and believe me, I've tried them all." (available for 386 or 486, DOS or Unix)

"We bought **INTERACTER** as a graphics library for PC , Unix and VAX, but now we use it for menus, input screens and system access too. INTERACTER makes our code user-friendly ***and*** portable."

"We started with 150,000 lines of vintage FORTRAN... **SPAG** translated it to beautifully structured Fortran 77 which worked first time. The whole operation took 2 man days; we estimated 6 man *months* without SPAG. ***That's*** what ***I*** call productivity"

STOP PRESS The *plus***FORT** toolkit (which includes SPAG) has been greatly enhanced. New features include a global static analyser, standardisation of declarations, automatic clutter removal, variable renaming and many more.

Call today for a free demo diskette featuring these (and other) products.

If you use FORTRAN, you MUST check this one out.

Polyhedron Software Ltd.
Linden House, 93 High Str. Tel. (44) 0865-300579
Standlake, Witney OXON Fax. (44) 0865-300232
OX8 7RH United Kingdom Compuserve 100013,461
US TOLL-FREE FAX 1-800-777-5519

Polyhedron Software

CIRCLE NO. 365

INNOVATIVE DESIGN TOOLS

FROM GREAT WESTERN INSTRUMENTS

Great Western Instruments are the specialists in innovative design tools for the embedded systems developer. Our services include full technical and design consultancy support for all our products.

NEW PRODUCTS

EMBEDDED DOS - **Fully MS-DOS 3.31 compatible. Optimised for real-time with multitasking, reentrant. For PC and embedded targets. Adaption kit includes source code for device drivers etc. £455. Circle 380**

DEVICE DRIVER SDK - **Full information and example source code for writing MS-DOS device drivers. Compatible with MS-DOS, DR-DOS and EMBEDDED DOS. £175. Circle 381**

DEMO DOS - **Royalty free MS-DOS 3.31 compatible for demonstration disks. Bootable. Make your demo more reliable - not influenced by host configuration. £195 Circle 382**

OMFCVT - **Use Turbo Debugger or DEBUG/RT to debug Intel iC-86 or PL/M-86 code for embedded applications. Converts OMF-86 to Turbo Debugger format. £380. Circle 383**

OVER-C+ - **Compact non-preemptive scheduler with message queues, triggers, timers and memory management. Supports Borland C++ and MS C. £345. Circle 384**

INSIGHT - **Debug tool gives detailed information about operation of AMX kernel. Works with Turbo Debugger on PC or via remote link. £395. Circle 385**

PARADIGM DEBUG/RT - Customised versions of Borland's Turbo Debugger for embedded product design. Displays special function registers, extended instructions for 80186, 80C186EB, V25, V25+, V40 and V53. Also supports Intel iC-86 and PL/M-86. **£255** **Circle 386**

AMX 86/286/EU000/Z80 - Real time executives from KADAK. Fast, reliable, compact and ROMable. Small code size. Preemptive task scheduler. Many facilities. C, C++ and assembler support. From **£995** **Circle 387**

PARADIGM LOCATE - Locator fully supports Borland C++ and MS C. Complete with DOS emulation and FP support. Many examples and start up source code supplied. TD, DEBUG/RT Emulator (OMF-86) and EPROM programmer support. **£345** **Circle 388**

PARADIGM TDREM - Target based monitor links with Turbo Debugger or DEBUG/RT via RS232 or parallel link. Supports 80x86, V-series. Uses only 5k ROM and 2k RAM. Many serial driver sources supplied and full custom information. **£195** **Circle 389**

PERSONAL SELECT - World renowned Low-cost Real Time pc based CASE tools supporting Yourdon/Hatley/Ward Mellor or HOOD. Easy to use. Single user **£495** Project-Select for multi-users over LAN **£995** **Circle 390**

SMS - Fully featured software management system. Easy to use. Version Control, Change Management, Configuration Management and Modification Request Management. Available for MSDOS, OS/2, OS-9, UNIX, VMS platforms. From **£490** for single user MSDOS. **Circle 391**

The above is only a fraction of the full range of design tools available. Call John Legg for further details.

GREAT WESTERN
- Understanding the needs of other designers !

For Immediate information about these & other products in our range, use your Fax/Telephone to call **Fax-direct** (0865) 727232, **Item 444401**

Unit 1m Farrington Fields
Farrington Gurney
Bristol BS18 5UU
Tel 0761 452116
Fax 0761 453226

VAT Extra, Carriage Free

All trademarks acknowledged

pared using SE350 and SL.EXE, a librarian program, allows you to add and remove entries from SIF files.

However, SIF files really come into their own when used in combination with the speech driver SPEECHDR.BIN. This file can be loaded in CONFIG.SYS to create a SPEECH device. Next, a SIF file is loaded as a TSR using the utility LIBINST. Now programs can be given the power of speech simply by writing to the SPEECH device, for example in BASIC:

```
OPEN "O" #1,"SPEECH"
PRINT #1, "A B C D E F"
PRINT #1, 26 'Send EOF
CLOSE #1
```

An alternative system, based on a single TSR, allows the programmer to load and play speech without altering CONFIG.SYS - obviously desirable for commercial programs. Quantech charges a modest licence fee for applications incorporating the SoftSpeak sound.

I should say that the sound-playing programs did exhibit a tendency to crash or lock the keyboard, especially on machines that deviated from the norm in some way. Network cards, low memory, DR DOS and QEMM all seemed to cause problems - occasionally. A SoftSpeak sample was included with the .EXE survey program, which was run on over 500 different machines. We had about 8-10 technical support calls apparently relating to sound problems, which doesn't seem too bad, given the heavily hardware-dependent nature of the package. And the problems *could* have been caused by *my* software, of course.

I have not enough space to detail all the other SoftSpeak goodies. There is a 'compiler', which converts .SOF files into self-playing .EXEs. There is a compression utility, to allow you to store large recordings. There are numerous example sound files, players, alternative recording programs etc. As well as the PCs own speaker, the package supports the Covox 'Speech-Thing'; Quantech itself offers various upgrade hardware for boosting output. There is extensive on-line help, and a clear and witty manual. There's lots more, but I must move on to the other package.

Monologue Sound drivers

Speak without a speech accessory
Hearsay 100, 500, or 1000
Sound Blaster (Creative Labs)
Echo PC+, MC, or Echo 1000
IBM Speech card
IBM ACPA (Audio Capture & Playback)
Tandy SL/TL/RL or Tandy 2500
Covox "Speech Thing" accessory
PS/1 Audio Card & Joystick Card

Figure 2 - The Monologue Sound drivers

Monologue

First Byte's Monologue is also a PC speech program, but of a very different nature. SoftSpeak plays back pre-recorded sound. Its speech, however distorted, started life in a human larynx. Monologue's speech is entirely synthetic; each consonant and vowel sound is constructed separately according to pre-determined rules, then complete words are assembled from the components.

The system, which arrives on two 360 KB disks, is blessed with the most fussy INSTALL program I have ever encountered, ever ever. 'Which drive are you installing from? Are you sure? Which drive are you installing to? Are you sure? Do you wish to install to the hard disk? Are you sure?' You think I'm making this up, but I'm not. Monologue occupied 250 KB on the hard disk - less than the capacity of the distribution disks because it only installs one driver. See Figure 2 for a complete list of sound devices supported. I was also asked whether I wanted to install Low Level, Mid Level or High Level speech. Which you select is a determined by the hardware - especially clock speed - which you are going to use to run Monologue. The only way to find out is by trying all the options - and each time you have to re-install the whole package, which seems a bit hard.

The main application itself is a TSR called MONO which occupies about 45 KB of conventional RAM and 130 KB of EMS/XMS. Operation is the essence of simplicity. You bring up Monologue over the text-based application of your choice with Alt-T. You mark a block, either using cursor keys or by using a mouse to click on the anchor points of your selected rectangle. Hit ENTER and Monologue then speaks the highlighted text.

Inevitably, Monologue sounds like a depressed adenoidal dalek with an American accent. Various configuration options allow you to configure the thing as, eg, a *female* DADWAAA, which is an improvement on machines with particularly feeble internal speakers - the higher frequencies are better able to penetrate the roar of the power supply fan. To give credit where it's due, Monologue does better than a young child learning to read: The voice is not a monotone - it goes up and down in pitch for individual syllables and takes account of punctuation. Since it doesn't 'know' what it is saying, it inevitably gets the stress of the words all wrong, which can make it difficult to understand. But, considering the problems involved, this is a good effort.

Monologue deduces the pronunciation of words from their spelling. This being English, this approach frequently doesn't work. To this end, First Byte supplies a look-up table (termed an 'Exception Dictionary') to which you can add extra words - for example requiring 'ocean' to be pronounced 'ohshun'. Unfortunately, I couldn't get this to work at all on my machine - the program always locked up when recording a new dictionary entry.

TALKDRVR.SYS allows you to add synthetic speech to programs - it operates in exactly the same way as SoftSpeak's SPEECHDR.BIN, except that the device created is called @TALK, not SPEECH. However, there is no equivalent to SoftSpeak's interrupt-hooked TSR. The add-on Speech Toolkit, which is not reviewed here, would presumably supply the extra functionality.

As with SoftSpeak, and presumably for the same reasons, Monologue was distinctly picky about the environment in which it ran - it crashed my machine at the slightest provocation. It proved a much more hardy animal on some of the other PCs in the office - this seems to be a luck of the draw thing. The manual states that the software will not tolerate multi-tasking software such as Windows, DESQview etc.

Conclusion

Although I have treated these packages as fun things, they evidently both have a wealth of serious applications - the most obvious in the area of making PCs useful to the visually handicapped. I am not going to name either piece of software as 'better' than the other because their function is so different. Choose SoftSpeak for reproduction and manipulation of recorded sound; if you wish to speak unknown words go for Monologue.

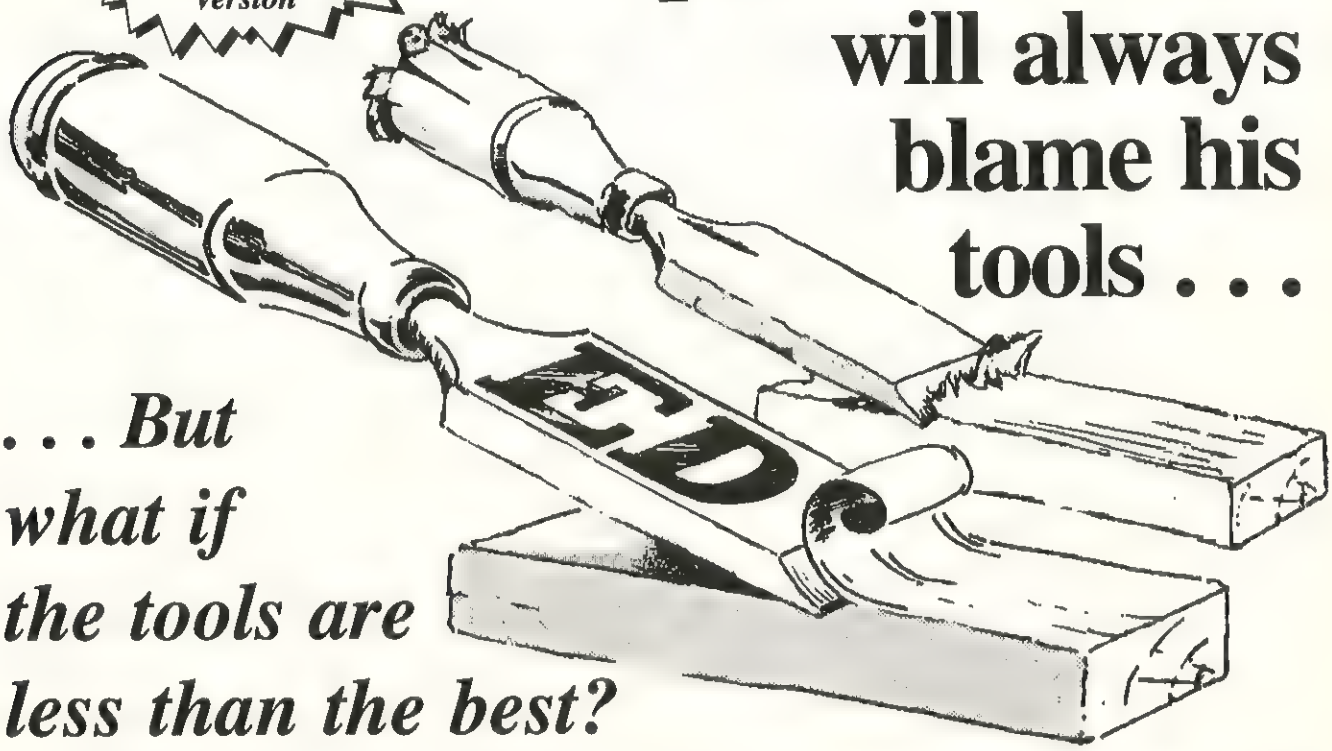
.EXE

SoftSpeak II+ costs £59.95 (software only) or £139.90 (with recording hardware). You can get it direct from the manufacturer Quantech (091 2280513), and we hear that it appears in the .EXE directory. Monologue is available from UK distributor Iansyst Ltd (071 6075844), price £89.

Aidan Ruff, co-designer of SoftSpeak, can be found sharing some of his sound expertise on this month's Code Page.

Ask about our NEW Windows 3 Version

A poor workman will always blame his tools . . .



. . . But what if the tools are less than the best?

A craftsman is only as good as his tools and a good craftsman always uses the best tools for the job. ED provides a rich, powerful text editing environment with all the features you need for programming in the 90s.

Ease of use, congfigurability and attention to detail are paramount in ED's design. You will be up and running right out of the box. ED even emulates other editors, Brief, Wordstar, Norton editor and Qedit . . .

ED's capabilities leave all other text editors behind and coming from QBS you get a level of service and support nobody else provides.

If you currently use cumbersome, slow, old fashioned tools, then it's time for a change. Take control, move into the fast lane with ED today, you owe it to yourself . . . . **ONLY £99**

Call for a free DEMONSTRATION DISC

Some of ED's Highlights

- Edit files to 100MBytes+, with lines up to 1024 characters.
- Use screen resolutions of 132 x 60 or more.
- Multiple windows and files.
- Cut and Paste text between files and windows with just a few keystrokes.
- Full support for rectangular Column blocks.
- Unlimited ***UNDO*** and ***REDO***.
- Easy to use pull down menus, context sensitive help, help bars and quick reference guides.
- Full ***LAN*** and multiuser support.
- Run compilers with error tracking.
- Releases all memory for your compiler.
- Language sensitive editing for ***C/C++***, ***dBase***, ***Clipper***, ***DataFlex***, ***PFX+***, ***Pascal*** etc.
- Fast ***C extension language*** for ultimate control. Includes source code for nearly all editor commands.
- Regular expression search and replace.
- Smart indenting, templates, object matching, code completion etc.
- Named keyboard macros.
- Flexible, fast, easy to use and fully configurable.
- Calculator, Calendar, Line drawing, ASCII chart etc.
- Save/restore editor state between sessions, including files and windows.
- Clipboard, Text buffers and Bookmarks.
- Run any DOS program or Shell to DOS.
- Flexible printer support and background printing.
- Directory Tree.
- Word wrap and Paragraph reformat.

Plus much more.

Ed
the programmer's editor

Soft as it Gets

QBS Software Limited,

10 Barley Mow Passage,
London W4 4PH

Tel: 081-994 4842
Fax: 081-994 3441
BBS: 081-747 1979

"Thanks for a great product. It's just what I've been looking for in an editor. I have tried many editors in my time, including Wordstar, Brief, Norton Editor, VIPC but none match ED's power and flexibility."
David Bagnara, Bell Organisation Pty Ltd, Victoria

"I have deleted Brief from my machine." *Damian Lewis, Kenneth A. Hansen & Associates P/L, Victoria*

"Brilliant editing system."
Geoffrey Evelyn, Tyne & Wear

Big Blue's OS/2 kit

In among IBM's vast catalogue of software and hardware are hidden one or two surprising items. Michael Price investigates the PS/2 Tools for OS/2 Developer's Kit.

The prosaically named *Developer's Kit 1* is part of a series of PS/2 Tools for users, developers and administrators of OS/2 systems. The PS/Tools kits contains programs and utilities originally developed by IBM for internal use. The Developer's Kit 1 contains a variety of programs to assist OS/2 developers and technical support personnel and also provides examples of OS/2 PM graphical effects. The most striking program is the system and application debugging tool, ASDT.

Before exploring the Developer's Kit 1, it is perhaps worth mentioning briefly the other software in this series; not least because all the kits are £62 each, with subsequent licenses to copy at £31 each (the Editor was astonished that IBM chose to sell *anything* for less than £100). *User's Kit 1* is a personal productivity package with PMDIARY. The user can keep track of dates and activities using diary, calendar, todo list, note pad, mini-database, mini-spreadsheet and pop-up reminders. There is also a copy of the Loadram2 program (described below), more recent than the version in Developer's kit 1. *User's Kit 2* adds more utilities, to locate files and check for duplicates, a more advanced calculator, a useful 'point and shoot' manager, a file browse and a disk space display. These utilities may be of interest to developers as well as to ordinary users. *Additional Developer's* and *Administrator's* kits are being planned.

The kits have no manual or fancy packaging, just a single diskette. All documentation is provided in .HLP (OS/2 help) and .INF (information presentation facility) formats. The OS/2 VIEW command can be used to search the .INF documentation in hypertext fashion or to print the sections. The diskette is packed using the Loadram2 utility, and Install and expand programs are also provided.

AS	ASCII	M7-M9	Set Variables M7-M9
CA	Colour Attributes	NW	Next Window
CB	Complex Breakpoint	OB	Output Byte
CG	Configuration	OW	Output Word
CM	Compare Memory	PD	Print Disassembly
CO	Code Origin	PI	Program Identification
CP	Copy Memory	PM	Print Memory
CT	Step Count	PR	Print Screen
DA	Disassemble ASM86	RC	Retrieve Command
DK	Define Key	RI	Restore Interrupt command
DM	Disassemble MASM (Intel)	RK	Reset Keyboard Hot Key
DW	Disassembly Window	RT	Resume Thread
D1-D9	Select Display	SB	Sticky Breakpoint
EB	EBCDIC	SC	Screen
EP	Execute Profile	SK	Set Keyboard Hot Key
EX	Execute	SR	System Reset
F	Find	ST	Step
FA	Find ASCII	S1-S9	Set Breakpoints S1-S9
FC	Find CSECT	TI	Take Interrupt Command
FE	Find EBCDIC	TP	Terminate Process
FX	Find Hex	TS	Task State Register
HP	Help	T0	Suspend/Reactivate 80386 Hardware Debug Registers
HT	Halt Thread	T1-T4	Set 80386 Hardware Debug Registers
IB	Input Byte	UM	User Mask
IP	Instruction Pointer	U1-U9	User-defined Functions
IW	Input Word	VW	IDT Window
LC	Location Counter	V1-V9	Set Variables V1-V9
LD	LDT Register	WA	Window Assumption
LW	LDT/GDT Window	XS	Extra Selector Register
L1-L9	Alter Window Area		
MW	Memory Window		
M1-M6	Alter Window Area		

Figure 1 - ASDT commands

Developer's utilities

The Developer's kit includes six utilities on the diskette, as well as the ASDT tool discussed below. These include LOADRAM2 (a file packer/unpacker), OS2LOCK (a keyboard lock), PSPM (a program that displays status on running processes), VIEWCLIP (a utility to display the PM clipboard) and PMSRCH (for file searching on disk drives, including networked driver).

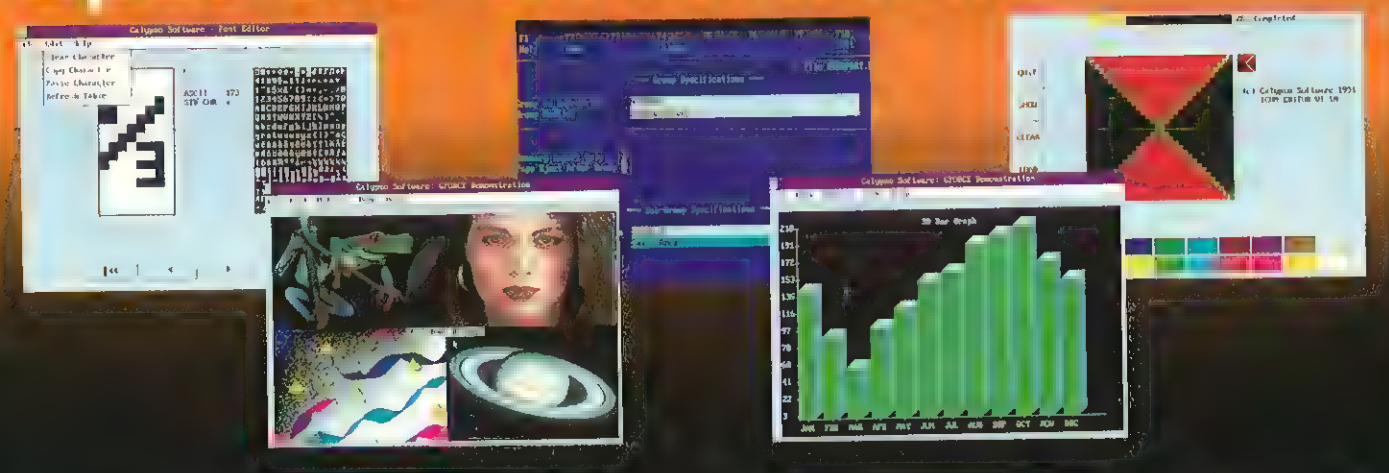
ASDT

ASDT is without doubt the most impressive program on the diskette. It is a full-screen assembly-level debugger that can be used for any OS/2 program, including device drivers and applications that run in either the protected or real mode or both. If you think of DOS DEBUG, and add the capability to deal with all the complications of the OS/2 environment (multiple chip modes, multi-tasking etc) then you have a rough idea of the role filled by ASDT. It's something of a low-level hacker's tool - for example there is no symbolic support.

ASDT gains control from OS/2 via Int 1, Int 3, NMI, or a hot key. Additionally, it is invoked on certain error conditions (Int 0, Int 6, Int C and Int D). Step execution (single, multiple or procedure) is supported and you can stop program execution at up to 162 specified points. You can display, alter or assign processor registers, search, compare or copy memory, and disassemble blocks of instructions.

ASDT uses a full screen format to display breakpoints, registers, memory, disassembled instructions, and other information.

Spot the difference...



All of these screens have been taken from Clipper applications. The difference is that four of the screens use the GFORCE graphical user interface.
GFORCE provides a very fast Windows™ like environment as a linkable library for the Clipper compiler.
GFORCE includes low level functions for pixels, lines, boxes, circles, fills, bitblits and high level functions for 3D buttons, pull down menus, dialogs, scroll bars, icons, bitmaps and fonts.
GFORCE retails for £185.00 including full documentation, Norton Guides, Quick library with complete Clipper source, graphical font and icon editors with Clipper source and ninety days technical support.

GFORCE works with Clipper 5.01, Summer '87 and overlay linkers such as Blinker™ etc.

Please call for a free demo disk.

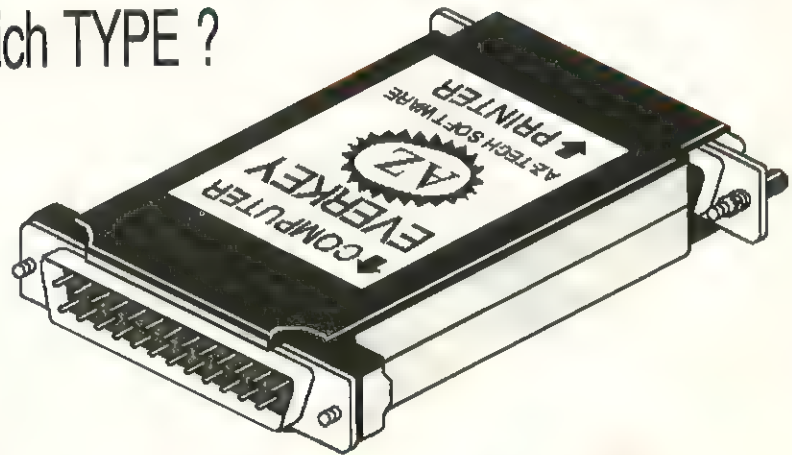
QBS Software Ltd
10 Barley Mow Passage
London W4 4PH, ENGLAND
Tel: 081 994 4842
Fax: 081 994 3441

GFORCE™ Clipper Graphical User Interface

CIRCLE NO. 367

LOOK at all OPTIONS.

Software Protection - but which TYPE ?



- **EVERKEY II** — *The Hard ware Option?*
  SINGLE WIRE ZERO LOAD - Total Compatibility.
  Not available in ANY other Key product.
- **EVERLOCK** — *The Soft ware Option?*
  Software-Only Copy Protection; economical & SECURE.
- **EVERTRAK** — *Non Copy-Protection Option?*
  Non Copy-Protection - Anti Hacker software protection.

CALL FOR A FREE WORKING DEMO DISKETTE

Other Services include:- Disk Duplication & supply - Label Production & supply - Packaging.

Sole Distributor for Az-Tech Software.inc - UK, Ireland & Spain

..... THE ONLY OPTION.

USER FRIENDLY
MicroSystems
22A, Bartleet Road,
Washford Industrial Estate,
Redditch, Worcestershire,
B98 0DG. England.
Tel: 0527 510 105
Fax: 0527 514 229

CIRCLE NO. 368

You can use multiple screens or a remote terminal to view both your application screen and the ASDT screen at the same time, and use the 80386 hardware debug registers to trap instruction execution or specific memory reads or writes.

Despite the 'PS/2 Tools' title, ASDT runs on a PC/AT as well as the PS/2 under OS/2 1.1 and later, and uses approximately 122 KB of memory. It runs as a device driver, in both the protected (virtual) mode and the DOS compatibility box (real) mode.

Starting up

To install ASDT, use LoadRam2 (supplied on the diskette) to unpack the ASDT.RAM file into a directory included in the PATH and LIBPATH statements in your CONFIG.SYS. Then add two device statements to your CONFIG.SYS file for the ASDT.EXE and the ASDT2.EXE programs. The first statement can include a profile of ASDT commands and options to set the hot-key scan code, use multiple screens, or use a remote terminal. If you want to debug another device driver during its initialization phase, you must make sure that ASDT is installed prior to that device driver in your CONFIG.SYS file.

To start ASDT and take control from OS/2, run the program you wish to debug making sure it contains an Int 3, usually as the first instruction. You can single step, executing one instruction in your program, using the ST command. Alternatively, the SN command sets one of the breakpoints to an address in your program. Then the EX key executes your program until it terminates, reaches a breakpoint, or reaches an ASDT-captured interrupt.

The main usage guidance comes from the ASDT.INF file, which is in effect the on-line manual. Of course, you have to switch away from ASDT to an OS/2 window in order to VIEW this document.

To return to OS/2, use the TP (Terminate Process) key. In protected mode, TP is valid only for processes that are not running at level 0. For those processes that are, you need to execute (EX) to completion to return to OS/2. In the DOS compatibility box, you can use the TP command only to terminate an application (not a device driver, a resident interrupt handler etc).

Display

The ASDT display is full screen, and can be switched between disassembly, memory and descriptor layouts, with nine separate sets of displays supported.

The disassembly display shows parts of memory as hex and mnemonic instructions. The memory display is standard IBM, ie it shows addresses and contents as sixteen bytes (one paragraph) in hex format, and the same sixteen bytes in either ASCII or EBCDIC format. You can scroll and alter memory locations in either of these displays.

The LDT/GDT display shows LDT and GDT descriptors, with the table index, descriptor type, and other descriptor information depending on descriptor type. Similarly the IDT display shows IDT descriptors. These displays are browsing only and do not allow you to edit the descriptor table entries.

Each form of the display includes the command line that is used to enter ASDT commands (see Figure 1). The commands are separated with semicolons and as many can be placed on the command line as will physically fit. There is no restriction on the sequence or combinations of commands. An alternative to typing ASDT commands on the command line is reading them in from a profile on disk. The profile is read in at ASDT installation time, and may be executed as required using the EP (execute profile) command.

ASDT provides permanent, complex and procedure step breakpoints. On each of the nine ASDT display screens, up to 18 permanent or complex breakpoints, or 17 such breakpoints and one procedure step breakpoint, can be specified. All breakpoints that are set are maintained across all processes and threads that are running. You can deactivate a breakpoint that is set for one process or thread while ASDT has control over some other process or thread. Similarly, you can reactivate a breakpoint that belongs to another process or thread.

ASDT can be forced to stop on every occurrence of a breakpoint by using the SB command. This sticky behaviour is usually reserved for physical address breakpoints and GDT virtual address breakpoints.

ASDT allows you to set the 80386 hardware debug registers (assuming, of course, that you are running on an 80386). The debug registers support both instruction and data breakpoints and can pinpoint when a data item (for example) is altered.

Using ASDT

A useful debugging technique is to use a different ASDT display (D1-D9) for each program (process) or code segment (CSECT or thread) being debugged. You can switch from screen to screen and can also save the contents of one screen into another. On each screen, the CO, breakpoints, disassembly locations and memory locations are unique. The registers and flags are the same across screens.

When an EX or ST command is executed, and the processor is subsequently stopped by a breakpoint, the ASDT display screen containing that breakpoint is automatically displayed. If the processor is stopped by something other than a breakpoint, the ASDT display that was active at the time of the EX or ST remains active.

Each of the nine displays has nine V breakpoints and nine S breakpoints available, which provides a total of 162 possible breakpoints. The S breakpoints are turned on when set, but the V breakpoints are not turned on until set on with a command. This gives the V breakpoints visible scratch pad functionality.

ASDT can be moved to the secondary terminal (if you have two adapters attached).

```
ASDT - IBM OS/2 DEBUG TOOL REL 1.4  06/20/90          DISASM   ASCII    D1
V1:.....  2:.....  3:.....  4:.....  5:.....  6:.....  7:.....  8:.....  9:.....
S1:.....  2:.....  3:.....  4:.....  5:.....  6:.....  7:.....  8:.....  9:.....
AX:  0000   BX:  5534   CX:  0010   DX:  0300   FL:  2246          Stack +0: 0BD8
BP:  4F38                SI:  001B   DI:  0000   [x0100010 01x0x1x0]     +2: 32FA
SS:  003B   CS:  01F8    DS:  0BD8   ES:  0BD8     NPLODIT SZ A P C       +4: 2E2A
SP:  F3E0   IP:  2C9B    CO:  ....:....              STEP CT: 0001        +6: 0BD8
            EX:004C1B    LC:  ....  TIW     0A78(DS),0009       OP: 0BD8:0A78=0000
==> u2 sc
                                                TR:          ....:....-....:....
L1: *  01F8:2C9B  F706780A0900    TIW     0A78(DS),0009              0BD8:0A78=0000
L2:    01F8:2CA1  741D            JE      01F8:2CC0                  01F8:2CC0
L3:    01F8:2CA3  50              PUSH    AX
L4:    01F8:2CA4  0F01E0          SMSW    AX
L5:    01F8:2CA7  D1E8            SRL     AX,1
L6:    01F8:2CA9  7305            JNL     01F8:2CB0                  01F8:2CB0
L7:    01F8:2CAB  E81400          CALL    01F8:2CC2                  01F8:2CC2
L8:    01F8:2CAE  EB0F            J       01F8:2CBF                  01F8:2CBF
L9:    01F8:2CB0  06              PUSH    ES
M1:    01F8:2CB1  E8E3EB          CALL    01F8:1897                  01F8:1897
M2:    01F8:2CB4  68D80B          PUSHI   0BD8
M3:    01F8:2CB7  1F              POP     DS
M4:    01F8:2CB8  E80700          CALL    01F8:2CC2                  01F8:2CC2
M5:    01F8:2CBB  E8EFEA          CALL    01F8:17AD                  01F8:17AD
M6:    01F8:2CBE  07              POP     ES
```

Figure 2 - The ASDT display

The Programmer's Interface

A range of products to aid the serious programmer. Libraries, tools and utilities to improve your productivity. Devote your time to developing better applications, not re-writing proven software, already available at low cost.

Human Interface Manager £295.00
A large of library functions for developing user interfaces. Produces portable code for a variety of platforms.

C Tools Plus £95.00
A library of high quality functions to enhance Microsoft C. Complete with source code and examples.

Dynamic Memory Control £69.00
Complete memory management. A great tool for configuring and tuning bespoke systems.

4C Classic £89.00
Hypertext editing for all C compilers. Full-featured editor, code analyzer and much more. Superb time saver.

Bar Code Library £189.00
Comprehensive library to aid fast and easy generation of bar codes. Full source code available.

Slate £189.00
Universal printer driver library. Provides a printer interface independent of the printer being used.

PC-Lint £79.00
Absolutely vital utility to check the quality of your software. Save hours in wasted debugging effort.

Power C Compiler £29.00
Just one of a host of superb quality low-cost C products from Mix. Other similarly priced programs are available.

BEAT THE STOCKMARKET
FREE copy of Blueberry Data's share analysis program,PRICE TRACKER, (advertised at £150) with every account worth over £200, excluding VAT

Prices do not include VAT but do include UK delivery.

These are just a few products from our range. Call for a complete list. We are able to offer full details on all products and advice on selection.

Blueberry Data Ltd
67 Hoppers Road
London N21 3LP

Tel : (081) 886 1319
Fax : (081) 882 9606
Mon-Fri : 9am - 10pm
Sat-Sun : 9am - 6pm

Call for best price on any technical software product.

CIRCLE NO. 369

KIBWORTH COMPUTER TRAINING

Language courses are normally held in groups of up to three and are tailored to suit individual needs:

C — *For structured and procedural programming (and inline assembler code).*

C++ — *A better C: Object-oriented programming and design*

Windows — *Various ways of implementing event-driven GUI's, including Windows 3.*

DATABASE — *Relational databases: PARADOX with PAL and the Paradox Engine*

Kibworth village is on the A6 between Market Harborough and Leicester, easily recognised by road and one hour by InterCity from London. Local hotels afford excellent value.

For further details contact Maude Richards.

Kibworth Computer Training
68 Springfield Crescent
Kibworth Beauchamp
Leicester LE8 0LH

CIRCLE NO. 370

Development speed = Oregon C++

Oregon C++ is, quite simply, the fastest C++ Development System software you can buy. Consisting of a true optimising compiler (not just a C translator), a source level debugger and libraries, it sets speed records in every area. Here's how.

The optimising compiler means optimum speed

Because the optimising compiler directly generates compact object code it eliminates the translating step. The result? Faster compilation, direct debugging and faster program development. And you end up with smaller and faster applications. What's more the compiler is switch selectable for C++, ANSI C, or K&R C.

You don't waste time worrying about compatibility

Oregon C++ conforms to ANSI standards for the C++ language and supports inter-language calls to and from C, Pascal, Modula-2 and Fortran. So you get fast, easy, access to existing code modules, without wasting time rewriting or re-debugging. And Oregon C++ is totally compatible with all existing C libraries.

Fast debugging is done in the same language in which the application is written

The Oregon Debugger - ODB - debugs C++, ANSI C and K&R C in the original application language, which means you get more reliable code in less time. And because the ODB understands multiple inheritance, it can quickly display the class hierarchy.

Oregon C++ is fast and easy to use

With a choice of command line or mouse-driven window user interfaces, Oregon C++ is easy to use. In window mode you even get separate windows for the application and debugger I/O, and every window is fully scrolling so you can find the information you want in no time at all.

Fast information...

For an instant response to your request for information, call us today. Or clip the coupon for your free technical data.

Instrumatic®

The Pan-European Technology Group
Instrumatic UK Ltd.
First Ave., Globe Park, Marlow, Bucks. SL7 1YA
Tel:0628 476741 Fax. 0628 474440
Tlx 847042 IMATIC G

For more information fill in the coupon and post today!

Name
Position
Company
Postcode Telephone

CIRCLE NO. 371

Similarly, it can be moved to a remote terminal (if you have one connected via COM1 or COM2). In either case, the application will be displayed on the main monitor.

ASDT can be used on application (as opposed to development) machines to catch elusive errors like Trap Ds. However, note that other non-affected processes will be halted as well.

User-defined functions

Up to nine user-defined functions can be called from within ASDT. When ASDT gives control to a user-defined function, the routine runs at level 3 and can call OS/2 I/O service routines that ASDT cannot call from level 0. All the user-defined routines (U1-U9) must reside in the dynamic load library, UX.DLL. You can create your own UX.DLL, or substitute your own functions for the one that comes with ASDT. Using the supplied libraries, for example, the ASDT screen may be saved to disk with the ASDT command 'U2 SC'. This approach was used to obtain the screen snapshot shown in Figure 2.

To activate ASDT from your application, you would normally write a tiny assembler routine with an INT 3, followed by a RET, and call that routine from your program. Some compilers, such as Microsoft C 6, allow you to add in-line assembler code. In these cases the INT 3 instruction can be placed directly in the application source code. You might also check that ASDT is installed before you actually issue the INT 3, perhaps by using DOSOPEN against ASDTDRV. Note, however, that if you are debugging an OS/2 Device Driver, DOSOPEN is valid only in the device driver initialisation routine.

Other debuggers

ASDT will work with CodeView or other application-level debuggers that use DosPTrace to achieve debugging. You will need to disable ASDT's use of GPF and other interrupts. This can be carried out in the CONFIG.SYS device statements for ASDT, or dynamically from the ASDT command line.

Some developers use the MS SDK kernel level debugger (KLD). This is a line-mode debugger which requires a remote terminal and keyboard connection. It does have the advantage that it formats many OS/2-specific data structures, and it has symbolic support. However, ASDT works with the standard retail version of OS/2.

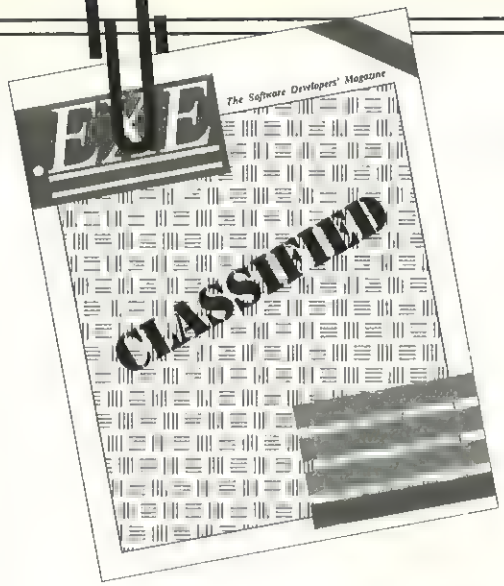
Summary

Even when OS/2 2.0 becomes available (currently predicted to be next spring) ASDT will continue to be useful for OS/2 1.x implementations and 80286 machines. It is to be hoped however that ASDT, along with other utilities on the PS/2 Tools diskettes, will find their way into the new package. In the meantime, the PS/2 Tools offer useful functions at quite reasonable cost.

EXE

Michael has worked in the mainframe environment for many years, in technical and development roles, and switched in the early eighties to the architecture and design of systems based on PCs and local area networks. Currently, Michael is a systems design consultant in the financial industry arena.

The PS/2 Tools were provided for review by IBM UK Limited (081 747 0747).



FOR YOUR EYES ONLY

To:..... All readers

From: ..... Agent .EXE

Re:..... CLASSIFIED LEAKS

Undercover sources inform of .EXE's intention to include classified advertising from February 1[illegible]... contents as yet unspecified.... expect adverts for shareware, utilities, hardware, games etc.....

The Competition are worried that this will offer unrivalled opportunities for readers and existing advertisers to reach thousands of Software Developers at minimal cost, without recourse to large display adverts.

Costs are anticipated to be as low as £85 per advert.

Our informant also tells of special offer available to existing readers of .EXE who have products to advertise.....

Make contact with Marc Warren on 081 994 6477 quoting password "extension 2238" for more inside information.

THIS INFORMATION IS CLASSIFIED.

Looking forward looking back

Jules explores why cyberspace is still bubbling under after 20 years.

It was back in the early 90s that virtual reality became actual. In only six months, three different and incompatible systems appeared; after about 18 months the market settled down, and the two competing systems we know today emerged. Both of these systems are now made by about a dozen manufacturers, which is a very odd, considering total sales of these systems last year totalled only ECU 20M! [Indat 0-23681-23-576] Surely the whole idea is dead, so why don't people let it lie down? What is so important about it that companies are queuing up to lose money in it?

The term 'cyberspace' was coined in the early 80s by William Gibson, an uninspiring science fiction author who wrote a series of books around his idea. [Glib 123-51513] Few people have heard of him now - his writing echoed the pathos of his time, and as the world's problems changed so the 'cyberpunk' style became less and less comprehensible.

At about the same time, NASA and the US Air Force were experimenting with head-mounted displays for controlling aircraft and feeding information to pilots. As technology developed the cost of such systems fell, and the quality of the images improved, until in 1992 it was possible to buy a basic toy system for about ECU 15,000. This was not a bit like today's systems - it was severely limited in its image quality, it used only polygons for modelling, and it had only 5 frames per second refresh. [Indat 0-23681-23-576-34] In spite of this atrocious performance, the press went wild! Over 4000 magazine articles were written, 300 TV documentaries were broadcast across Europe, and 'cyberspace parties' lasting for days at a time were 'happened'. Within a very short time, an entire culture had grown up around this technology, with its own patterns of behaviour, its own values, and its own vernacular. [Glib 1254632]

At the time, the common perception was that we could both work and play in cyberspace. Both the technology and the language suggested it was a real place - putting on the head-set, for example, was called 'popping in'. The migrainous images, and the appalling quality were all seen as problems which would sort themselves out in due course, as computers became even faster.

The main problem, I believe, was that cyberspace was (and still is, in many ways) barely possible technology (or BPT). When the machines were first made they were so clever that people just wanted an excuse to use them, and technology was proceeding at such a pace that the common perception claimed that everything is possible if enough money is thrown into it. Nevertheless, it is interesting that the only people at the time who seemed to have any idea what the technology could be used for were hippies - the people who had a history of filtering the real world out of their experience.

In the event, it became very hard to find anything to use this stuff for. There were tasks which were inherently three-dimensional (such as chemical engineering or certain CAD problems) but it transpired that the solutions found in cyberspace were often impractical to manufacture, and as soon as designers could work in 3D they wanted more degrees of freedom still - particularly the chemists wanted to know more than what shape something was; they wanted to see it reacting with complex cocktails. Indeed, the blind spots in drugs engineering in cyberspace became abundantly clear after the Ceopan disaster in 1998. [Indat 0-25239-65-46]

It was recognised early on that, in order for the technology to achieve the critical mass required to make it cheap and generally available, a mass-market use had to be devised - in effect, it was seen by its proponents to be a solution looking for a problem! Business use had driven the proliferation of computers in the 70s and 80s, so business was again tapped for applications. Entire virtual offices were constructed, with virtual filing cabinets, virtual typewriters, and virtual desks. What nobody expected was that, while these were fine places for doing virtual work, real work was utterly impractical! In fact, a real office worked better, and the advantages conferred by putting people in different countries into the same room were hardly measurable.

And herein lies the problem. Even after twenty years of development, reality is still better at its job than a computer is. No computer can model the rich relationships that the real world has, and even the quality we can put into a machine is far in excess of telecommunications bandwidth. These things just can't talk to each other fast enough.

Most surprising of all is that a related, but far more powerful and far cheaper technology has been completely ignored - telepresence. [Indat 0-138254-5346-5] Telepresent systems formed the core of flight simulators in the 1950s and 60s, and have been used for orbital engineering for years (although the time lags mean that operating the systems is not really like 'being there'). On the other hand, the current breed of chip robots are being driven by exactly this technology, and the advances we have seen in real-space surgery in the last five years are only the beginning of what is possible. [Indat 2-3638654-45]

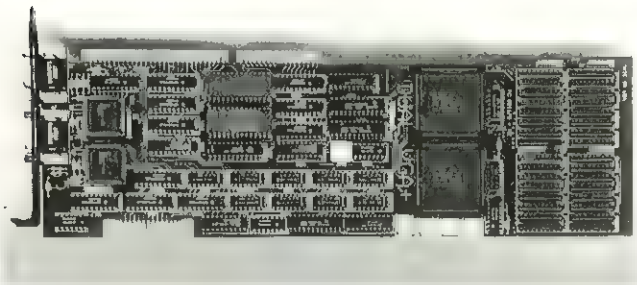
So why do companies still invest crippling research budgets into cyberspace? I suspect it is the same reason the motor companies used to build high-performance racing cars - to understand the technology, and to filter the lessons learned down to real systems. Cyberspace represents a problem that will probably never be solved completely, so as such is an ideal medium for blue-sky research.

EXE

Jules May has been a regular contributor to .EXE for over twenty years, and used to be a consultant before he became a talk show host. He can be contacted on CIX as `jules`*, and still keeps his phone number (0707 44185) for sentimental reasons.*

MULTIPLE INDEPENDENT MONITORS
CAN NOW BE DRIVEN SIMULTANEOUSLY
FROM A SINGLE PC OR PS/2

THE DUAL VGA PLUS



★ UP TO 800×600 RESOLUTION
★ TWO INDEPENDENT SIMULTANEOUS DISPLAYS WITH OVERLAY CAPABILITY
★ 100% COMPATIBLE WITH IBM VGA/EGA/CGA/MDA VIDEO STANDARDS
★ 4 BOARDS IN ONE SYSTEM GIVE 8 SIMULTANEOUS DISPLAYS

NEW DRIVER! ALLOWS WINDOWS 3.0 TO BE EXPANDED ACROSS FOUR MONITORS

CEBRA COMMUNICATIONS LIMITED
26 LORNE PARK ROAD, BOURNMOUTH
TEL 0202 299048 FAX 0202 299192

CIRCLE NO. 374

Before You Leap™

A superb range of Intelligent Software Estimating Tools

- Function Point Analysis *(IBM/Albrecht)*
- MKII Function Point Analysis *(Nolan Norton/Symons)*
- MKII FP Estimating Method *(Nolan Norton/Symons)*
- COCOMO - Constructive Cost Model *(Boehm)*
- Import Function Points from CASE tools *IEW/System Architect/Excellerator....*
- Export to Project Management tools *PMW/Timeline/SuperProject/MS Project....*
- Risk Analysis
- Full Life Cycle estimates/Maintenance
- Schedule compression
- Staff resourcing/Cash forcasts
- SSADM V4.0 compatible
- High Performance and ***proven*** accuracy
- Calibrates to user environment
- Superb Colour Graphics

DOS/WINDOWS 3/NETWORKS

"Hands On" Demos £25 - Full systems from £595
Written in the U.K. and available throughout Europe

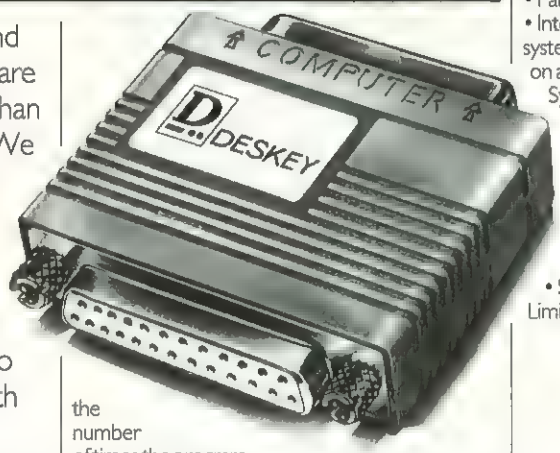
Strategic Systems Technology

14 Landons Close
Jamestown Harbour
Prestons Road
London E14 9QQ
Phone: (071) 538 8228
Fax: (071) 515 3887

CIRCLE NO. 375

'DESkey' Software Protection system from Data Encryption Systems Limited

We have more experience and expertise in the design of software protection modules (dongles) than any other company in the UK. We sell only products designed in house by our own engineers – the same engineers that give you technical support. More than 13 years experience in the design of dongles have gone into our current product range with features such as:



- Pseudo Random Number Generator. Billions and billions of random numbers without repeating. Software and Data encryption could not be easier. 'Seedable' too!
- Through Encryption. Data may be fed into the DESkey for on-line encryption. This keeps the encryption key hidden.
- Memory. Up to 240 bytes of memory split into 'Public' and 'Private' sectors. The 'Public' sector may be read from and written to at will. The 'Private' sector may be read at will, but for writing, it requires your customer specific password.
- Down Counter. You program into the DESkey the number of times the program should run before stopping. This gives you the ability to sell 'goes' of your software rather than an open licence to use it forever – or even to send a fully working demonstration copy that will stop working after say, 10 goes.
- Variable Response Algorithm. This feature is similar to the well known 'public-key' algorithm and works in conjunction with an algorithm on the host computer. Makes any attempt at software emulation impossible.
- Secure Memory Read. Even if the same memory data is read repeatedly, the data returned from the DESkey never appears the same – this also makes the DESkey impossible to emulate.
- DESlock® automatic .EXE or .COM file encryption system. No need for source or .OBJ files. Takes only 5 seconds or less to protect your software.
- Transparent and Cascadable/Stackable. Does not interfere with any other device wanting to use the same port - even allows other manufacturers dongles to work!
- Parallel, Serial or Bus versions available.
- Intelligent Serial devices suitable for any operating system such as Unix, Xenix, OS/2, DOS etc. Works on any hardware including PC Networks, Mini Systems or even Mainframes.
- No programming units required and NO hidden extras.
- Free lifetime telephone support direct with the designers.
- Free evaluation Kits.
- Guaranteed Exclusivity to all our customers.
- Fast Order Turn-round.
- Sold only through Data Encryption Systems Limited.

Data Encryption Systems Limited

Edbrook House, Cannington,
Bridgwater, Somerset. TA5 2QE
Telephone (0278) 653456
Fax (0278) 653300

CIRCLE NO. 376

PCL - Speed is not enough

The first programming language written specifically for the PC has not yet shaken the world. Peter Flynn found it fast but in need of some further thought.

Many of you may remember the adverts for PCL (Personal Computer Language), which started appearing a few years ago. In a flush of enthusiasm I shelled out the £100 or so requested, because some of the facilities mentioned seemed to place it well above most other languages at that price. *(Editor's note: Unfortunately, we became aware at the last minute that Peter's copy of PCL is about two years old. However most of the Third Side issues - language design and syntax etc - remain unaltered. For details of the current PCL, please contact the supplier via the phone number given at the end of this article.)*

The package is neat: a single disk and a manual. Installation is a straightforward `COPY` into a directory of your choice. I just about managed that, then settled down with a coffee to browse the manual. It seems a bit mean to carp so soon, but the typography is not good: the text is set with no contrasting typeface for examples of code, so actually making sense of some of it can be quite difficult, and it is very tiring on the eye. Because of this, persevering meant I had to read hard, so I actually learned more than you usually do from a first run at a manual (maybe it's deliberate after all?).

Iterations	Test	BASICA	MS-BASIC	TURBO	PCL
10000	Empty loop	4.6	.23	.25	.04
1	BYTE Magazine Calculation benchmark	252.2	17.32	31.88	14.82
	With 8087	-	-	6.29	6.09
1	Displaying 24 lines of 80 characters on screen	5.0	4.05	2.77	.15
1	Formatting & displaying 100 decimals	6.95	2.02	1.51	.42
	With 8087	-	-	1.35	.25
1000	Converting decimals to character string	22.0	3.56	3.89	3.17
	With 8087	-	-	2.87	1.04
1000	Converting character strings to decimals	50.7	3.92	8.18	2.43
	With 8087	-	-	3.48	1.12
1000	Catenate 2 character strings of length 10	3.2	.65	.58	.32
1000	String search	4.79	.99	1.03	.28
1000	Sequential write 90 byte records	18.3	9.0	7.5	2.9
1000	Sequential read 90 byte records	18.2	8.6	7.4	2.5

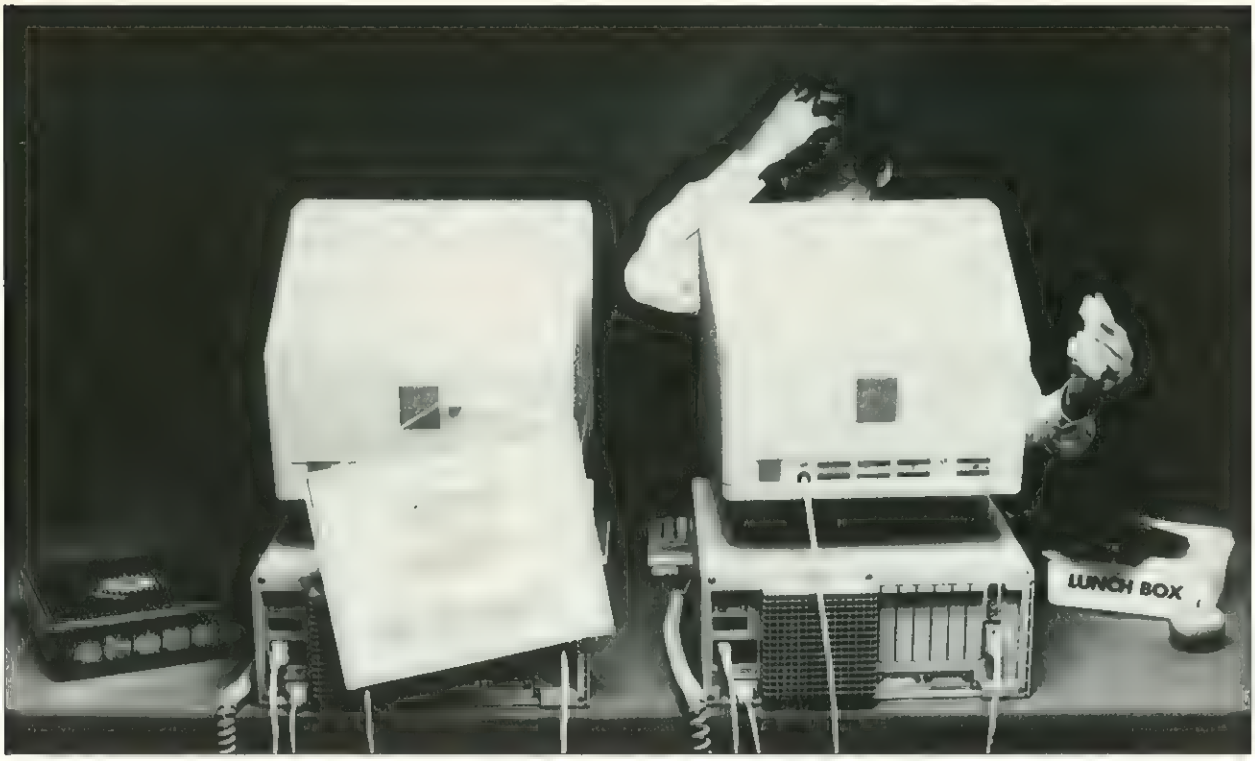
All timings are in seconds. They were taken on a standard IBM PC with PC-DOS 3.1, a real-time clock and a 10 MB hard disk. Where decimal numbers were involved, the highest precision available was chosen for each compiler.
BASICA is the interpreter distributed with PC DOS 3.1. The compilers used were Microsoft Basic 2.0 and Turbo Pascal 3.0.

Figure 1 - Benchmark performance figures claimed by PCL

PCL is a bit like a cross between BASIC and C, in that you have some pretty good bare-metal control of the computer while keeping syntactical and verbal simplicity. Thus you have a whole chapter at the end on machine-level commands (preceded by a warning not to tinker) including interrupt-handling and register access as well as a lot of assembler; but you can also write plain old data-processing programs with a minimum of punctuation, using as good an approximation of English as any other language.

The author, one Wolfgang Lilienfeld, tried hard to keep what I (and presumably he) perceive as the pettiness of other languages at bay. For example, there is no need for data conversion, as everything on the right-hand side of an assignment is automatically converted to the correct type for the left-hand side, with sensible defaults. (Scrambling sound as all supporters of strongly-typed languages leave the lecture theatre). Equally, there is no need for parentheses around arguments of functions; the structural logic of the command is used to determine what is an argument and what is not.

Or nearly. I found that complex expressions involving functions of more than one type (eg arithmetic manipulation of the integer position of a character within a string) meant you do need the parentheses. This doesn't bother me, but a novice might find it a little off-putting. What does bother me is that the parentheses are required for delimiting but not for changing precedence.



Sycero from System C. The best thing on the programmer's menu

Sycero is a powerful 4GL and applications generator for dBASE, Clipper or C programmers.

Sycero is powerful enough to create even the most sophisticated database applications. In fact, it gives you the best of both worlds: the speed and ease of use of a 4GL with the flexibility of a 3GL.

Sycero's complete development environment consists of a data dictionary, screen and report painters, a powerful 4th generation language based on the dBASE/Clipper standard. It also includes automated program types such as menus and file maintenance, reducing the amount of programming needed for common tasks. Networking code can be automatically generated and high quality documentation is produced as standard.

The multi-file report generator lets you build the simplest of lists or the most complex invoice, and automates sub-totals, pagination, report loops, user selection of ranges and sort sequences and wildcard matching.

So, if you want to stay ahead in PC database development and **still** have time for lunch, call System C now on 0622 691616, or return the coupon below.

System

System C Ltd 60-61 High St Maidstone Kent ME14 1SR
Tel 0622 691616 Fax 0622 691241

To: System C Ltd 60-61 High St Maidstone
Kent ME14 1SR
Please send me further details about your Clipper generator, Sycero dB ☐
Please send me further details about your C generator, Sycero C ☐
Name .....
Company .....
Address .....
.....
.....Postcode .....
Tel .....Fax .....

CIRCLE NO. 377

dBASE is a trademark of Ashton-Tate. Clipper is a trademark of Nantucket.

The only precedence (says the manual) is strict left-to-right evaluation.

Implementation

PCL can be made memory-resident, which is a sensible idea for a language that does not come with an editor or tools: you can then use your own favourite editor to write the code, and a double-shift invokes the system. However, there's no way to pass the file name to it, so you have to save your file and then give the RUN command followed by your filename when PCL pops up.

Or nearly. Actually PCL is not a compiler but a kind of interpreter, that is, it does not produce a relocatable object file on disk. Which means, of course, you cannot link it to make a .EXE or .COM file. What it does provide is lightning-fast interpretation into non-relocatable execute code, which you can save but not link. This code is also very fast, so fast I had difficulty believing my first trivial program had done anything! Figure 1 shows some manufacturer's data from the manual underlining this speediness.

The drawback of not being able to make executable DOS programs for distribution is probably a major reason why PCL has not caught on. I find extreme difficulty in understanding the reasoning, but a clue is in the manual, early on, where the author discusses the absence of another important language feature, pixel graphics:

'... due to the grotesque inefficiency of the standard colour card and the absence of any common standard for the others. Rather than program round incompetent hardware, the PCL design team decided to omit pixel graphics altogether, and not to implement the usual inefficient and unsatisfactory solution, which would not have matched the PCL standard in all the other areas.'

Now, whereas it is probably uncontroversial to state that the PC's graphic standard is unsatisfactory (especially given that PCL was designed before VGA became so prevalent); I think it's been a pig's breakfast from the beginning. But it seems a bit pointless for a software author to take umbrage at the design of the machine for which he is writing, to the point of omitting useful features from his programs. Perhaps this attitude is showing through in other areas.

Either way, if you want to distribute software written in PCL, you have to ship your source code plus a copy of the PCL interpreter itself (presumably in breach of copyright?) which is plumb crazy. The ASCII file containing the manual upgrade to Version 3.07B makes it clear that saved images may not be loaded and run on another machine (natch, if they're full of absolute memory references, no-one else has a machine configured like yours). Certainly the system would have done better if it could compile object code that can be linked.

Facilities

Having said all that, the language itself has plenty to recommend it, and is full of time-saving features. There are no reserved words; it has automatic co-processor detection, very fast I/O, no limits on file size (apart from your hardware), and built-in access to DOS file attributes, volume labels etc.

For full-screen applications, there is a set of commands for logical record data entry, with definable screen fields including validation which can be invoked as a group, giving a level of control over record entry with cursor control, field colours and function key support which is very fast to program.

Window management is another nice feature. You can define a window anywhere on the screen, saving the background for a later restore, just with two commands. There is no indication of any explicit support for other windowing systems (notably DESQView because it's character-oriented like PCL, rather than GUIs like MS Windows), although PCL certainly runs within DESQview without any problems.

File-handling is unusual in that the OPEN function is automatic and implicit on the first use of a READ: there is no OPEN statement. Control over formatted reading/writing is excellent in both binary and ASCII modes, but (unlike the comparable xBASE languages) there is no support for indexed access: you have to code and maintain your pointers by yourself. No problems for computer science weenies who always write their own routines anyway, but anyone who has ever been forced to do this will be aware of the pitfalls for the novice.

Loop control shows a similar attitude: you have to define and increment your own counters, as the DO statement merely establishes and performs the loop: it has no facility for counter control other than a simple integer iteration limit. You can use WHILE to control loops, but the counter control, especially for non-integer incrementing, is your own business.

```
! Displays a sorted directory for any path, any drive.
! Hit Scroll Lock to prevent display from scrolling.

CHAR    DIRECTORY[300,45],FN,DIR
INTEGER N

        IF FN='' then  FN=?SUBDIR ()+'*.*'
        PROMPT 'File name : ',FN,30;

        WINDOW 2,1,21,80,7; CLW
        IF ?ESCAPE>0 THEN STOP

        DIR=?DIR FN; N=0; DIM DIRECTORY
        WHILE LEN DIR>0 THEN DO
             INC N; DIRECTORY[N]=DIR
             DIR=?DIR
        ENDDO
        IF N>0 THEN DO
           DIM   DIRECTORY,N
           SORT  DIRECTORY
           ? DIRECTORY; COLOUR 12; IF ?ESCAPE>0 THEN STOP
           ? 'Total size ';TAB 14;FORMAT '*********'; ? FSIZE (FN),','
        ENDDO
        FORMAT ''; ? N,' entries'; COLOUR 7
```

Figure 2 - PCL program to display a sorted directory listing

```
! Solution to the Third Side problem in PCL
float a,b,c
prompt "Enter the first side:",a
prompt "Enter the second side:",b
prompt "Enter the third side:",c
? a,b,c," is "
if a+b<c | a+c<b | b+c<a then ? "not a triangle";
   else if a=b & b=c then ? "equilateral";
   else if a=b | b=c | a=c then ? "isosceles";
   else ? "scalene"
wait;quit
```

Figure 3 - The Third Side problem

THE SOFTWARE CONSTRUCTION COMPANY

BORLAND

Borland C++*	£189
C++/App. Framework*	£259
** FREE UPGRADE TO 3.0 WHEN AVAILABLE*	
Turbo C++ 2nd. Edition	£45
Turbo C++/Turbo Vision	£65
Turbo C++ for Windows	£75
Turbo Debugger & Tools	£85
Turbo Pascal v.6.0	£65
Pascal Professional	£129
Pascal for Windows	£99
ObjectVision	£79
Paradox 3.5	£325
Paradox Engine	£189

UK AUTHORISED

DATABASES

Advanced Revelation	£795
Advanced Revelation - LAN	£695
askSam	£255
askSam LAN	£579
Clarion Personal Dev. 2.0	£79
Clarion Professional Dev.	£679
DataEase	£485
DataEase - LAN 3-PAK	£485
Dataflex	£449
Dataflex Multiuser	£1065
Knowledgeman/2	£645
Magic PC	£325
Open Access III	£449
Open Access III Compiler	£385
R:BASE 3.1	£515
R:BASE Compiler	£190
R:BASE 3.0 LAN Pack	£645

DBASE

Clipper 5.01	£325
dBASE III PLUS	£395
dBASE IV	£495
dBASE IV Developer's Ed.	£695
dBFast/DOS	£255
dBFast/PLUS	£255
dBXL	£165
dBXL LAN	£389
Force	£449
FoxBASE+	£295
FoxBASE+/386	£425
FoxPro	£495
Palcom "Paradox Compiler"	£319
Palcom Network Version	£645
Vulcan	£320

OBJECT-ORIENTATED

Smalltalk/V	£65
Smalltalk/V 286	£129
Smalltalk/V Windows	£325

CASE TOOLS

EasyCASE Plus 3.0	£320
EasyCASE Plus Prof. 3.0	£420
Smart Case	£195
System Developer I	£325

VERSION CONTROL

MKS RCS 5.1	£165
MKS RCS 5.1 - 5 User	£645
PVCS Professional	£319
PVCS - 5 User	£1345
SMS	£395
Sourcerer's Apprentice	£175
Sourcerer's Apprentice	£429

TEXT AND UNIX TOOLS

MKS AWK	£65
MKS AWK DOS & OS/2	£129
MKS Prog. Platform	£429
MKS Toolkit 3.2 *NEW*	£165
PolyAWK Toolkit	£145
SpellCode	£65

TRANSLATORS

BASTOC	£515
BAS_C Commercial	£579
CodeTranslator *NEW*	£129
FOR_C	£359
Pascal to C Translator	£99

EDITORS

BRIEF for DOS & OS/2	£199
BRIEF upgrade	£60
BRIEF with dBRIEF	£275
BTags	£35
Cheetah	£159

Microsoft

BASIC PDS 7.1	£199
Visual BASIC	£85
C 6.0 Compiler	£199
C 6.0 & Windows SDK	£299
COBOL 4.50	£389
FORTRAN 5.1	£189
MASM 6.0	£65
MS-DOS 5 Upgrade	£49
Pascal Compiler	£149
QuickBASIC	£49
QuickC	£49
QuickC/Assembler	£75
QuickC for Windows *NEW*	£85
QuickPascal	£49
Source Profiler	£39
Windows 3.0	£60
Windows S.D.K.	£199

U.K. Authorised

dBRIEF for dBASE/Clipper	£99
dBRIEF for Paradox	£99
dBRIEF for R:BASE	£99
EDT+	£190
EMACS	£210
Epsilon	£129
KEDIT	£99
MKS Vi	£99
Multi Edit Professional	£115
Norton Editor	£50
SPF/PC	£159
SYNDIE	£319
VEDIT PLUS	£120

LINKERS

.RTLink	£190
.RTLink/Plus	£319
ALINK	£129
Blinker	£175
Plink w/LTO	£359

MAKE UTILITIES

MKS Make	£99
MKS Make - 5 User	£320
PVCS Config. Builder	£129
PVCS Config. Builder/5 User	£485

PARSERS AND LINTS

MKS Lex & Yacc	£165
PC-Lint	£90

PROFILERS

.RTLink/Plus	£319
CHARGE	£85
Codesifter	£79
Inside!	£85
PC Metric	£129

DEMO BUILDERS

Dan Bricklin's Demo II	£190
Instant Replay Prof.	£385
Show Partner F/X	£259

MEMORY MANAGERS

386 MAX Version 5	£85
Above Disk	£79
Blue Max	£100
HEADROOM	£85
HEADROOM Network Extens	£39
HI386 Complete	£65
MOVE'EM	£59
QEMM-386	£69
QEMM-50/60	£65
QRAM	£55
Turbo EMS	£65

DOCUMENTATORS

4c for Brief	£79
ABC Flowcharter	£190
ASMFLOW	£129
C-Clearly	£85
C-Doc	£120
Clear for C	£129
Clear for dBASE	£129
dAnalyst for C and C++	£190
dAnalyst for Clipper	£190
dAnalyst for dBASE III	£190
dAnalyst for dBASE IV	£190
dAnalyst for FoxBASE+	£190
EasyFlow	£150
Flow Charting III	£165
SourceDoc	£190
The Documentor	£190
Tree Diagrammer	£65

DISASSEMBLERS

Sourcer 486	£85
Sourcer w/BIOS pre-proc.	£110

PROFESSIONAL SHAREWARE FOR DEVELOPERS

ASM Library (20)	£40
C/C++ Library (68)	£66
dBASE/Clipper (87)	£100
DOS Utilities (37)	£40
Novell Netware (54)	£66
Turbo Pascal (38)	£40
Visual BASIC (11)	£40
Windows (64)	£66

(Number of diskettes)

E-MAIL

cc:Mail	£450
cc:Mail EXPAND	£385
cc:Mail Remote	£190
DaVinci eMAIL	£645
DaVinci eMAIL Combo	£965

CODE GENERATORS

DATABOSS	£395
Layout	£195
ObjectVision for Windows	£259
PRO-C	£750
R&R Code Generator	£99
UI2 Developer's Release	£385
UI2 Touch & Go	£259
Zachary	£259

DEBUGGERS

Multiscope DOS	£115
Multiscope Windows	£245
Periscope 286/386 Pod	£389
Periscope I OK	£319
Periscope I/512K	£385
Periscope I/MC	£449
Periscope I/MC 512K	£449
Periscope II	£145
Periscope II-X	£129
Periscope IIw/Switch	£145
Periscope IV (16 MHz)	£775
Periscope IV (25 MHz)	£900
Sherlock	£129

MULTI-TASKING

DESQview 2.4	£89
DESQview 386	£159
Dr. Switch-OnCall	£65

BATCH ENHANCERS

Batcom	£39
Beyond.BAT	£65
Command Plus	£85
ProKey Plus	£85
The Builder	£99

TERMINAL EMULATORS

BLAST II	£195
COTERM/100	£145
COTERM/220	£160
COTERM/220	£160
COTERM/4010	£165
EM 320	£199
EM 4010	£230
EM 4105	£330
EM 4105 Network	£345
ICE.10	£190
SmarTerm 125	£190
SmarTerm 2392	£65
SmarTerm 240	£225
SmarTerm 320	£129
SmarTerm 340	£225
SmarTerm 400	£115
Smarterm 470	£190
ZSTEM 100	£65
ZSTEM 220	£99
ZSTEM 240	£190
ZSTEM 4014	£65

SCO UNIX

SCO Operating System-2	£525
SCO Operating System Unltd	£750
SCO Development System	£875
SCO Open Desktop - Pers.	£895
SCO Open Desktop Dev. Sys	£1295
SCO XSight Runtime	£350
SCO XSight Dev. Sys.	£350
SCO XSight Complete Sys.	£650
SCO Multiview	£395
SCO VP/ix - 2 user	£450
SCO VP/ix - Unlimited	£850
SCO FoxBase+ 286	£795
SCO FoxBase+ 386	£995

UPGRADES

App. Framework/Borland C++	£96
Turbo Vision for Turbo C++	£46
Turbo C to Borland C++	£100
Any Borland product to C++	£130

CALL FOR OTHER UPGRADES

*All prices exclude V.A.T. U.K. carriage included. ***One offer per order at published prices. Offer required must be stated at time of order.*** VISA or ACCESS accepted. Credit to approved accounts.

SUMMER '91 CATALOGUE

Call for your free copy. *PN Dec.*

1 THE MALTINGS, GREEN DRIFT, ROYSTON, HERTS. SG8 5DB
TELEPHONE (0763) 244114 FACSIMILE (0763) 244025

RS232-C (serial) operations are also included, even a glass-teletype terminal call, and it will also handle background tasks like running a screen clock while waiting for keypresses. There is a useful set of commands for examining the input buffer, so keypress control is possible to a very high degree, which I find important in writing utilities. The terminal subroutine itself seems to have some problems, though - I haven't been able to make it work anywhere.

Some idea of the compactness of PCL code can be gained from the sample routines supplied on the disk (Figure 2 contains a program to display a sorted directory); the fact that these run at a speed comparable to Turbo C or Lahey Fortran is a significant tribute to the way in which the language has been implemented.

Three-star programming

A solution to the Triangle problem is in Figure 3. This was very easy to write once I grasped the curious nature of the `IF` statement. Boolean `AND` and `OR` are used in PCL for bit-twiddling: in `IF`s you have to use `&` and `|` instead. The other quirk is that statements after an `IF` on the same line are not regarded as within the `TRUE` domain, as they are in BASIC.

In Figure 3, the whole nest from `if a+b<c` right down to `"scalene"` was all typed on a single line: it has only been broken up here for readability and to fit on the page. It would be possible to segment it into `IF...THEN DO...ENDDO` blocks, but that seemed pointless for this application, as there is only one action statement within the `TRUE` domain of each `IF`.

The language itself has plenty to recommend it - such a pity it doesn't compile to object files

`WAIT` causes a pause for a keypress: essential if you don't want the screen to blank out and return to DOS immediately on termination!

What would be interesting would be to try PCL out by converting a really robust large-scale application, such as TeX (which is a well-known compiler-breaker, used as such by many developers to prove or crack a language). If I find the time to try this one I will report back!

EXE

Peter Flynn is currently manager of the research and academic computing development service at University College, Cork. He is into early music, reading, surfing, typography and cyberspace. You can mail him as `pflynn` *on BIX and CIX, as* `pflynn@iruccvax.ucc.ie`*, or talk to him as* `silmaril` *on Relay, irc or cBix.*

PCL is produced by Calend (081 894 7409) and is available for DOS (£100 ex VAT) and OS/2 (£195 ex VAT).

System Science

• Software • Specialists • Software • Specialist • Software • Specialists • Specialist • Software • Specialists •

Unix

SCO Unix Op. Sys **SCO** Unix Dev Sys.
SCO Open Desktop **SCO** ODT Dev Sys
SCO TCP/IP & NFS FTP PC/TCP
Interactive Unix Op Sy / Architect Series
Informix 4GL & SQL **RM** Cobol-85
LPI Compilers **C Scape** Screen Lib
LPI C++ (native) 386 Unix & Sun- **NEW**
... many more for 386 Unix and Sun

Mathematics

Derive - The Mathematical Assistant
Mathematica 2 386/387 , MAC/ENH
Mathematica for Workstations (Sun, HP..)
MathCad for Windows (new) GrafTool
What's Best! **Lindo**
SPSS/PC, Statgraphics Chiwriter Sci WP

Windows Development

(see Microsoft and Borland, Zortech)

CASE:W Corp.	**£645.00**
Smalltalk/V Windows	**£295.00**
Btrieve for Windows	**£345.00**
Multiscope Windows Debugger	**£295.00**
Commonview	**£395.00**
Win++ by Blaise	**£150.00**

CrossDevelopment

2500AD Cross Assemblers	£135.00
Avocet Cross Assemblers	£call
Cross Compilers - Introl, Manx, Hitech	

Pascal (see Borland)

TopSpeed Pascal SE	**£115.00**
Object Professional	**£110.00**
Btree Filer MU	**£120.00**
Asynch Professional	**£89.00**
Blaise Turbo Vision Toolkit	**£89.00**
Blaise Turbo Asynch Plus	**£115.00**

Power Basic	**£89.00**
many Basic libraries for comms, graphics	
PC Logo	**£50.00**
LMI UR/Forth	**£295.00**
Smalltalk V/DOS	**£69.00**
muLISP 90	**£225.00**
RM Cobol-85 (new Dev Pack)	**£call.**
MS DOS 5.0, QEMM 386 etc	**£call.**

Microsoft

Quick C for Win **NEW!**	**£99.00**
Microsoft C 6.0	**£220.00**
MS Fortran 5.1 (new ver)	**£210.00**
MS Windows SDK	**£210.00**
MS Win SDK & C6.0	**£335.00**
MS Basic Compiler 7.1	**£225.00**
MS Cobol 4.5 (DOS & OS/2)	**£460.00**
MS Macro Assembler 6.0	**£79.00**
MS Visual Basic	**£99.00**
MS Quick C 2.5	**£60.00**
MS Quick C & Assembler	**£99.00**
MS Quick Basic	**£60.00**
MS Dos 5.0 upgrade	**£62.00**
MS Source Profiler (new)	**£49.00**

Authorised Languages Dealer

Borland

Borland C++ with AFW **NEW**	**£275.00**
Borland C++ (Dos & Windows)	**£219.00**
Turbo C++ Windows **NEW!**	**£call**
Turbo Pascal Windows	**£115.00**
Turbo C++ & Turbo Vision **NEW!**	**£75.00**
Turbo Pascal 6	**£75.00**
Turbo Pascal 6 Prof	**£145.00**
Turbo Debugger and Tools	**£95.00**
Paradox Engine	**£275.00**

Authorised Languages Dealer

Intersolv (Authorised)

PVCS for Dos	**£375.00**
PVCS Config Build (PolyMake)	**£165.00**
PVCS multi-user , OS/2 & Sun	**£call**
PVCS Professional Editor (SPE)	**£189.00**

MKS

MKS Toolkit new ver 3.2	**£155.00**
MKS RCS new ver 5.1	**£155.00**
MKS Lex & Yacc new ver 2.6	**£155.00**
MKS multi-user and OS/2	**£call**

Tools and Editors

Brief 3.1	**£195.00**
Kedit (Xedit for PC)	**£110.00**
.RTPatch for DOS, Win & OS/2	**£245.00**
.RTLink Plus	**£295.00**
Plink 86 Plus	**£295.00**
Personal Rexx new ver 3	**£110.00**
Ghost (auto software testing)	**£145.00**
Flowcharting III	**£150.00**
C Programmers Toolbox	**£155.00**

... and many more tools and utilities

C++ Compilers

Zortech C++ DOS & Windows	**£220.00**
Zortech C++ Dev Ed (incl OS/2)	**£345.00**
Zortech C++ Sci & Eng	**£525.00**
Zortech C++ Video (6xVHS)	**£275.00**
JPI TopSpeed C++ SE	**£115.00**
JPI TopSpeed C++ PE	**£175.00**
JPI TopSpeed C++ Pro Dev.	**£229.00**

C Compilers

see Microsoft

Watcom C 8.5	**£275.00**
Watcom C386 8.5 **NEW!**	**£495.00**
Aztec C 86 (with ROM support)	**£265.00**
TopSpeed C PE	**£175.00**

C Datafile

CodeBase 4.2	**£175.00**
CodeBase ++	**£175.00**
Ctree Plus	**£345.00**
Faicom Professional Tollbox	**£745.00**

C Communications

Essential Comms	**£175.00**
Greenleaf CommLib	**£175.00**
C Asynch Manager (Blaise)	**£115.00**

C Screens

CScape with Look & Feel	**£365.00**
Vermont Views 3	**£335.00**
Panel Plus II	**£275.00**
Zinc for C++ (Win &Dos)	**£225.00**

C Graphics

Essential Graphics	**£195.00**
PCX Programmers Toolkit	**£165.00**
Halo	**£195.00**
Metawindows	**£195.00**

... many more libraries

Pharlap 386 DOS Ext SDK	**£325.00**
Pharlap 286 DOS Ext SDK (new)	**£325.00**

Fortran Compilers

Lahey F77L	**£375.00**
Lahey F77L-EM/32 with OS386	**£875.00**
Salford FTN 77-386	**£765.00**
Salford FTN77-486	**£875.00**
PC/Interacter (screens)	**£325.00**
Ingraf (Sci - source)	**£195.00**

... many Fortran Libraries

- **Prices are exclusive of VAT.**
- **Prices include delivery to GB.**
- **Prices are subject to change.**
- **VISA, Access and Mastercard welcome with telephone orders.**

3-5 Cynthia St
London N1-9JF
Fax: 071 837 6411

071 833 1022

• Software • Specialists • Software • Specialist • Software • Specialists •

WE GUARANT
FIND A WIDE
SOFTWARE

"May I congratulate you on your contin
Software Paradise from all other n

WE'LL BEAT ANY NATION

Software Paradise specialises in Mail-Order Software for IBM Personal Computers (and the Apple Macintosh) running DOS, Windows™, OS/2® and UNIX/XENIX®. Over 15,000 software products from over 2000 individual manufacturers guarantees you the most comprehensive range of PC software in the world from any single source. With up to 50% discount on all the latest titles from the U.S. and the U.K., Quality Service, Huge Stocks, Next Day Delivery, Pre & Post-Sales Technical Support and "We'll BEAT Any Nationally Advertised Price" - There's no need to call anyone else. Software Paradise (established in 1986) regularly supply nearly all of the U.K.'s leading corporate and public sector organisations, regional health authorities, universities, colleges and local government departments as well as thousands of small businesses and private users. Our policy of "Service First" and total commitment to our customers in the supply of quality products and advice has earned us a very large and loyal customer base. "*The Discerning Person's Guide to Software*" is the most complete Buyers Guide to PC software available with over 200 pages of applications and development tools. Call us on (0222) 887521 or complete the coupon to receive your FREE copy.

"The prize for the flashiest catalogue in my news-box, however, goes hands-down to Software Paradise .... the 200 pages are chock-a-block with interesting utilities, specialist libraries, wacky compilers and obscure operating systems." EXE, July '91.

FREEPHONE 0800 378·873

ORDERLINE ONLY PLACE YOUR ORDERS FREE OF CHARGE FROM ANYWHERE IN THE UK

ALL OTHER ENQUIRIES TEL: (0222) 887521 (15 LINES) OR FAX: (0222) 862209 (3 LINES)

Software Paradise, Avenue House, King Edward Avenue, Caerphilly, Mid Glamorgan, CF8 1HE

PARADISE

WE SUPPORT THE ETHICAL USE OF SOFTWARE. TO REPORT COPYRIGHT VIOLATIONS CALL F.A.S.T. ON (0628) 660377 NEW! REACH US ON COMPUSERVE AT MAILBOX 100015, 3570

EE YOU WON'T
R RANGE OF
ANYWHERE †

d excellent service which distinguishes
order software suppliers." May '91

ALLY ADVERTISED PRICE*

What Makes Us Different

- ✓ FREEPHONE 0800 Orderlines
- ✓ FREEPOST Mailing Address
- ✓ 24-Hour FAX Orderlines
- ✓ Up to 50% Discount On Over 15,000 Software Titles
- ✓ Pre-Sales Consultation
- ✓ Post-Sales Technical Support
- ✓ Unbeatable Prices
- ✓ Next Day Delivery
- ✓ Huge Inventory
- ✓ Knowledgeable Sales Staff
- ✓ VISA, ACCESS & Mastercard Accepted
- ✓ No Credit Card Charges Until Shipping
- ✓ Corporate PO's Welcome
- ✓ Educational & Volume Discounts
- ✓ Latest Versions Guaranteed With Full Documentation & Manufacturers Warranties
- ✓ Over 6 Years Experience Providing The Very Best In U.K. and U.S. Software

Please Specify Media Size When Ordering
Goods Are Not Offered On A Trial Basis
All Prices Exclude Carriage & VAT
Prices & Specifications Subject To Change Without Notice

† The products you see listed in this advertisement represent a small sample of our full range

All Trademarks & Registered Trademarks Are Hereby Acknowledged

Software Paradise, Avenue House, King Edward Avenue, Caerphilly, Mid Glamorgan, CF8 1HE

* PRICE PROMISE

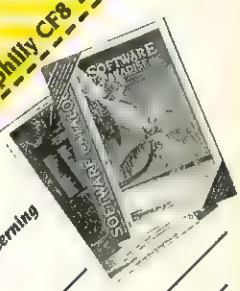
If you find an identical software product advertised in the national computer press by another mail order company based in the U.K. at a legitimate, current price - which is less than our advertised price - we will happily BEAT that price and SAVE YOU MONEY! We will also beat other suppliers with our friendly and efficient service and very wide range of brand-name, quality software. Since 1986 we've served 1000's of customers with this same promise. Discover today why we're one of the fastest growing companies in the industry. To all of our current customers - Thank you for your support, we really do appreciate it!

Access
MasterCard
VISA

Mail to: Software Paradise, FREEPOST, Caerphilly CF8 1ZZ

- FREEPOST TODAY -
NO STAMP NEEDED

Please rush me my FREE copy of "The Discerning Person's Guide to Software"



NAME
COMPANY
ADDRESS
POSTCODE
TEL & FAX

EXE 12/91

CIRCLE NO 393

PC Security from Microft Technology.

MENUGEN

MENUGEN provides a password protected menu interface. The system manager has complete control of what each user is permitted to do. Access to the DOS prompt can be prevented. MENUGEN has no memory overhead. **£48** + **VAT.**

CLAM

CLAM provides TOTAL security. It incorporates MENUGEN and D-LOCK. In addition it provides a number of other facilities, the most important of which is file encryption – the only way to provide a high level of security on a micro. **£148** + **VAT.**

D-LOCK

D-LOCK prevents access to the hard disk after booting from diskette. **£38** + **VAT.**

S-LOCK

S-LOCK clears the screen and locks the keyboard after a specified period of inactivity or on request. The screen is restored and the keyboard unlocked only when the correct password is input. **£38** + **VAT.**

Security Consultancy.

A professional computer security consultant can help you determine the steps required to protect your computer information and systems. The options offered range from a fixed price risk analysis with action advice to a full security audit.

Microft Technology Limited
The Old Powerhouse, Kew Gardens Station
Kew, Surrey TW9 3PS.
Telephone: 081-948 8255

CIRCLE NO. 394

Anti-virus software is also available. Corporate licences are available for all products.

The Sound of Software

Have you ever been puzzled over how some PC games programs manage to produce complex sounds without extra hardware? Aidan Ruff knows how.

What makes a good program great? Many elements. Ten years ago, the answer would have been 1) it works, 2) it is supported (to some degree), 3) there was a manual to get you past the flaky user interface etc. Now, however, these simple ingredients are no longer enough. We need **L**ights, **S**ound and **D**ramatic action - LSD for short.

Consider an area of programming into which nearly everybody delves periodically - GAMES. For a PC game to be good, it must incorporate all of the LSD ingredients. The graphics had better be good or you will have lost before starting. The story line and on-screen action will keep the player riveted, but even with the *L* and the *D* components in place, the magical *S* element is, in many cases, missing.

Ladies and Gentlemen, for your intellectual entertainment, I would like to launch into a description of how the PC produces its beeps and boops, followed closely by a piece of code to show you how to do it.

Sense and Compatibility

In modern PCs, the trend is towards large scale integration, where many elements which used to be implemented using standard ICs are now bunged into a few enormous chips, bristling with pins. In order to remain PC compatible, fortunately, the overall functionality of the hardware must remain the same as in the early days of IBM dominated personal computing. This has the consequence that some of the less known modes of operation of the various internal components of a PC are carried on through the various breeds of PC kind. This includes a device known as the CTC, or counter/timer chip.

A lot of the work involved in generating sound is done by the CTC. The CTC, sometimes known by its Intel part number, 8253, has three 16-bit counters that can be used in a number of useful ways. IBM had the foresight to connect the second of these three counters to a loud-speaker via a simple transistor buffer. Further, it allowed the output of the counter to be gated on or off by a port bit. I have a sneaking suspicion that this counter output was originally used to generate signals to record data onto cassettes, before it was realised that DOS doesn't do its stuff very well at 1200 BPS.

The remaining two counters in the 8253, numbers one and three, are used, respectively, to generate interrupts for memory refresh and maintain the system real time clock. I have managed to find a use for both counter two (sound generation) and counter three (timing control).

Incidentally, you can give your computer a small shot in the CPU - speed-wise - by increasing the count value in counter 1 and

Mode Number	Description	Possible use
Mode 0	Interrupt on terminal count. When the counter is loaded, the output goes low. When the count reaches zero, the output goes high.	Generating an interrupt after a programmed time has elapsed.
Mode 1	Programmable one-shot or monostable. Similar to mode zero, but a low to high transition of the gate input is needed to start the count.	Pulse stretching
Mode 2	Rate generator. The output will go low for one period of the input clock to the counter. A pulse on the output will occur every N cycles of the input clock, where N is the value programmed into the counter.	Divide by N counter. Generates a division of the input frequency.
Mode 3	Square wave generator. Most commonly used mode in PC software. The period of the output signal is equal to the input clock divided by the counter value.	Generating beeps and periodic interrupts.
Mode 4	Software triggered strobe. The output will go low for one input clock period after the terminal count has been reached. Re-initialised by reloading the counter.	Event delaying.
Mode 5	Same as mode 4, but triggered by the gate input by external hardware.	External event delays.

Figure 1 - Operating modes of the 8253

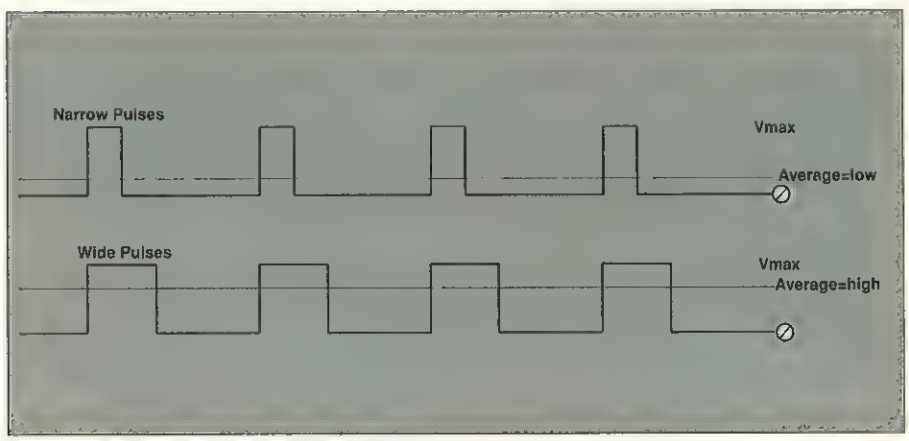


Figure 2 - Effect of varying the width of the pulses

```
; ****************************************
;
; This program sets up the PCs' real time
; clock interrupt to generate sounds.
;
; Non-commercial users are free to use this
; code as they wish. Commercial users may
; like to know that Mr Ruff enjoys crates of
; Newcastle Brown Ale very much.
;
; ****************************************
;       .model  tiny
        jumps
        locals
cseg    segment para public 'CODE'
        org     100h
        assume  cs:cseg,  ds:cseg, ss:cseg
firstbyte:
        jmp     main
; The sound driver program
;
; Functions are:
;
; ah = 0, speak a buffer in ds:dx,
;                 lngth in es:bx
; error in al,
;     0 = OK,
;     1 = buffer address not set
; time correction ratio
TIME_RATIO      equ  549
; counter to determine time correction
nbytes          dd   ?
; flag to indicate play in progress
playing         db   ?
; storage for old interrupt
oldint          dd   ?
; original status of port 61H
old61           db   ?
; period of note (frequency)
period          db   ?
; down counter for note generation
pcount          db   ?
; alternation flag for note generation
alternate       db   ?
; pointer to note sequence
bufaddr         dd   ?
; duration of note
duration        dw   ?
; amplitude of note
amplitude       db   ?
; the program's PSP address
psp_address     dw   ?
; signature for de-installation
signature       db   'SNDPLAYR',0

new_int proc    far
  cmp     ah, 0
; ds:dx points to a buffer
  je      @play_buf  ; speak a buffer
; don't recognise command
  mov     ax, -1
  iret

; **** play out the buffer in ds:dx ****
@play_buf:
  push    ds
  push    es
  push    di
  push    si
  push    bp

; Initialise the sound system, then wait
; for sounds to finish.
; Finally, remove sound system

  cli
  mov     ax,ds ; set up pointer to buffer
  mov     cs:[word ptr bufaddr+2],ax
  mov     cs:[word ptr bufaddr],dx
  mov     cs:byte ptr playing,1
; zero phase alternation flag
  mov     cs:alternate, 0
; now force a drop through intitialisation
; on the first interrupt
  mov     cs:pcount, 1
  mov     cs:duration, 1
; set up interrupts for sound playback
  mov     ax,3508h      ; function 35, int 8
  int     21h
  mov     word ptr cs:[oldint],bx
  mov     ax,es
  mov     word ptr cs:[oldint + 2],ax
  mov     dx, offset intr2 ; new interrupt 8
  mov     ax, cs
  mov     ds,ax
  mov     ax,2508h      ; function 25h, int 8
  int     21h
@set rate:
; now set up the interrupt rate and program
; CTC channel 3 for interrupt on terminal
; count
  in      al, 061h    ; read speaker port
  and     al, 0fch
                  ; store bottom 2 bits
  mov     byte ptr cs:[old61], al
  in      al, 061h
; enable speaker, set 8253, c2 gate on
  or      al,3
  out     061h,al
  ; (0b0h) set 8253 counter 2 mode
  ;  to programmable one-shot
  mov     al,0b0h
  out     043h,al
  ; set counter 0 to LSB/MSB load
  mov     al,34h
  out     43h,al
  mov     al,0
; set counter 2 to zero
  out     42h,al
  out     42h,al
  ; set counter 3 to LSB load only
  mov     al,090h
  out     43h,al
  mov     al,65 ; al <- interrupt rate
; set counter 0 to interrupt rate (65=10KHz)
  out     40h,al
  mov     al,0
  out     40h,al
  sti

; now wait for sound to finish
@waitend:
; still playing?
  test    byte ptr cs:[playing],1
  jnz     @waitend

  in      al, 61h   ; restore port 61 state
  and     al, 0fch
  or      al, byte ptr cs:[old61]
  out     61h, al
; now restore the old  int8
; ...interrupt rate already restored
; by 'intr2'
  ; store new interrupt 8 addr
  mov     dx,cs:[word ptr oldint]
  mov     ax, cs:[word ptr oldint+2]
  mov     ds,ax
  mov     ax,2508h
  int     21h
  pop     bp
  pop     si
  pop     di
  pop     es
  pop     ds
  xor     ax, ax
  iret
endp    new_int

;****************************************
; The sound generator routine
;
; Table driven parameters as follows:-
;
; <PERIOD of note (1 byte)>
; <DURATION of note (1 byte)
;                 = approx 1/80 sec>
; <AMPLITUDE of note (0-58, 1 byte)>
;
;****************************************

intr2       proc        far
  cli
  push    ds
  push    si
  push    ax
  ; are we playing any sounds?
  test    cs:[byte ptr playing],1
  jz      @tinc2
@carry_on:
  dec     cs:pcount
  jnz     @no_change
; change the phase alternation flag
  xor     cs:alternate, 1
  mov     al, cs:period
; reset counter for note period
  mov     cs:pcount, al
@no_change:
; check for which amplitude to output
  test    cs:[alternate], 1
  jz      @low
  mov     al, cs:[amplitude]
  jmp     @output_level
@low:
  mov     al, 1
@output_level:
  out     42h, al ; output the data byte
  dec     cs:duration    ; note finished?
  jnz     @byte_count
; now extract the next note (if any)
  ; read data pointer
  mov     si, cs:[word ptr bufaddr]
  mov     ax, cs:[word ptr bufaddr+2]
  mov     ds, ax
; * NOTE *
; this code does not allow music information
; to straddle segment boundaries
; move pointer on to next note
  add     word ptr cs:bufaddr, 4
  lodsb
  mov     cs:period, al ; store period
  lodsb
  xchg    ah, al
  lodsb
  xchg    ah, al
  ; store new note's duration
  mov     cs:[duration], ax
  or      ah, al        ; zero duration???
  jnz     @more_notes
  ; zero 'playing' flag
  mov     cs:[playing], 0
  ; slow interrupt rate down
  mov     al,0
  out     40h,al
  out     40h,al
  jmp     @nodec
@more_notes:
  lodsb
  ; store new note's amplitude
  mov     cs:[amplitude], al
; keep track of how many times the interrupt
; routine has been called so that the real
; time clock can be corrected later
@byte_count:
  inc     word ptr cs:[nbytes]  ; inc LSW
  test    word ptr cs:[nbytes], 0ffffh  ; 0?
  jnz     @nodec
  inc     word ptr cs:[nbytes+2] ; inc MSW
  jmp     @nodec
; increment the time
@tinc2:
  push    dx
  mov     ax,40h
  mov     ds,ax
  mov     si,06ch
  mov       ax,word ptr [si]  ; inc time
  mov       dx,word ptr [si+2]
  inc       ax
  or       ax,ax   ; wrapped over to zero?
  jnz     @notzero1
  inc       dx
; check for gone into next day
@@notzero1:
  cmp    dx,18h
  ja      @goneover1
  jb      @store1
  cmp       ax,0b0h
  jb      @store1
@goneover1:
  inc   byte ptr [si+4]    ; inc day
  xor       ax,ax       ; zero time
  xor       dx,dx
@@store1:
  mov     word ptr [si],ax
  mov       word ptr [si+2],dx
  pop       dx
@@nodec:
  mov     al,20h     ; send EOI to 8259
  out       20h,al
  pop       ax
  pop       si
  pop       ds
  sti
  iret
; finished saying current word, prepare for
; next word or terminate if no more words
@endw:
  mov     al,0  ; slow interrupt rate down
  out       40h,al
  out       40h,al
; now correct the time
  push      dx
  push      bx
  push      cx
  push      di
; read bytes spoken in last word
  mov     ax,word ptr cs:[nbytes]
  mov     dx,word ptr cs:[nbytes+2]
  mov       cx,TIME_RATIO
  div       cx
; store no of ints required into cx
  mov       cx,ax
  add       cx,13  ; time correction value
;
; update the time
  mov     ax,40h
  mov       ds,ax    ; clock base location
  mov       si,6ch
```

Figure 3 - SND.ASM sound driver TSR program *(Continued on page 74)*

so slowing down the memory refresh rate. This means that your machine will spend less time memory refreshing and more time processing. Typically, expect 5% speed increase, for free! Just imagine, an extra three minutes per hour to spend on coffee breaks.

Operating modes

There are six modes of operation for the 8253, these are outlined in Figure 1. Mode 3 is the one that most programmers aim for, because it generates standard bleeps and boops for minimal effort. There are few opportunities for special effects, however.

The mode which I find the most versatile is Mode 0. Imagine for a second that you are outputting a regular sequence of pulses of a fixed width. Varying the width of the pulses will vary their average energy. Wider pulses, more energy, narrower pulses less energy. (See Figure 2 for clarification.) This technique is commonly used in motor control circuitry. The motor averages out the variable pulse width and changes speed accordingly - this also overcomes start-up stickiness and low speed torque problems. Incidentally, it also reduces power output stage power dissipation, since the drive transistors are either on or off - ie maximum voltage drop, minimum current, or minimum voltage drop, maximum current. If you have the guts to open your PC, remove the speaker and fit a more beefy transistor, you can try this for yourself.

How does this help us? If we tie in the real time clock interrupt to a new routine which changes the timer count value between two values, say 1 and 100. Then we will generate a square wave output with a certain frequency and a certain energy. So what? Mode three will do the same thing, won't it? Yessss... it will. However, if we reduce that second value to, say, 50, we also reduce the average energy of the output signal. (That's the volume level to you, mate). Now, if the rate of change between the two values is *F*, then we are generating a signal of frequency *F/2*. Stupendous.

So, what do we have? Well, we have a way of generating - under interrupt - a sequence of variable pulses with variable average energy and if we don't change back and forth between our two values on every interrupt, but do it, say, every second interrupt, then we can generate a signal of frequency *F/4*. Extend this to every third interrupt and I think that will you start to get the idea... variable amplitude and variable frequency. Rock and roll!

Making Music

Now all we need is some way of interfacing this to an application. I have included a program in Figure 3, written using Borland's TASM, which will generate all of the sounds that you will need. You can, of course, pick out the salient bits and put them into your own code, but, unless you have a good understanding of interfacing assembly level code to your compiler, or are writing at that level, then the easiest way to access the sound system from *any* programming language is by using a TSR, accessed by software interrupts. (As Figure 4 is a TSR, you should make the final executable a .COM program, either by using EXE2BIN or setting the appropriate flag - /t with TLINK - on your linker.)

There are some interrupts that are generally left alone by DOS, these start at int 60H. Applications that require a software interrupt access point can usually take up 60H, 61H, 62H, 63H or 64H with impunity - so long as they are not already in use, of course. A well-behaved piece of code should pick a free interrupt and change its

```
  mov       ax,word ptr [si]; LSW
  add       ax,cx
  mov       word ptr [si],ax
; bx <- new LSW clock count
  mov       bx,ax
  mov       ax,word ptr [si+2]; MSW
  adc       ax,0
  mov       word ptr [si+2],ax
; check if we have gone over a day boundary
  cmp       ax,18h    ; check MSW
  ja     @incday2
  jb     @stotime2
  cmp       bx,0b0h   ; check LSW
  jb     @stotime2
@incday2:
  inc    byte ptr [si+4] ; inc day counter
; calculate extra time over day boundary
  sub       bx,0b0h
  sbb       ax,18h
@@stotime2:
  mov   word ptr [si],bx; store new time
  mov       word ptr [si+2],ax
  pop       di
  pop       cx
  pop       bx
  pop       dx
  jmp    @nodec
intr2       endp

lastbyte:         ; the end of the program
;                   ********************
; Everything past this point is throwaway
; when TSR is activated
start:
main    proc    near
  push  cs
  pop     ds
; first access the address of the PSP,
; to access the command line arguments
  mov     ah, 51h
  int     21h
  mov     es, bx          ; es <- PSP
  mov     cs:[psp_address], bx ; save PSP
  mov     di, 80h    ; addr of command tail
; al <- bytes in command tail
  mov     al, es:[di]
  or      al, al
  jz      @int_check5
  inc     di
```

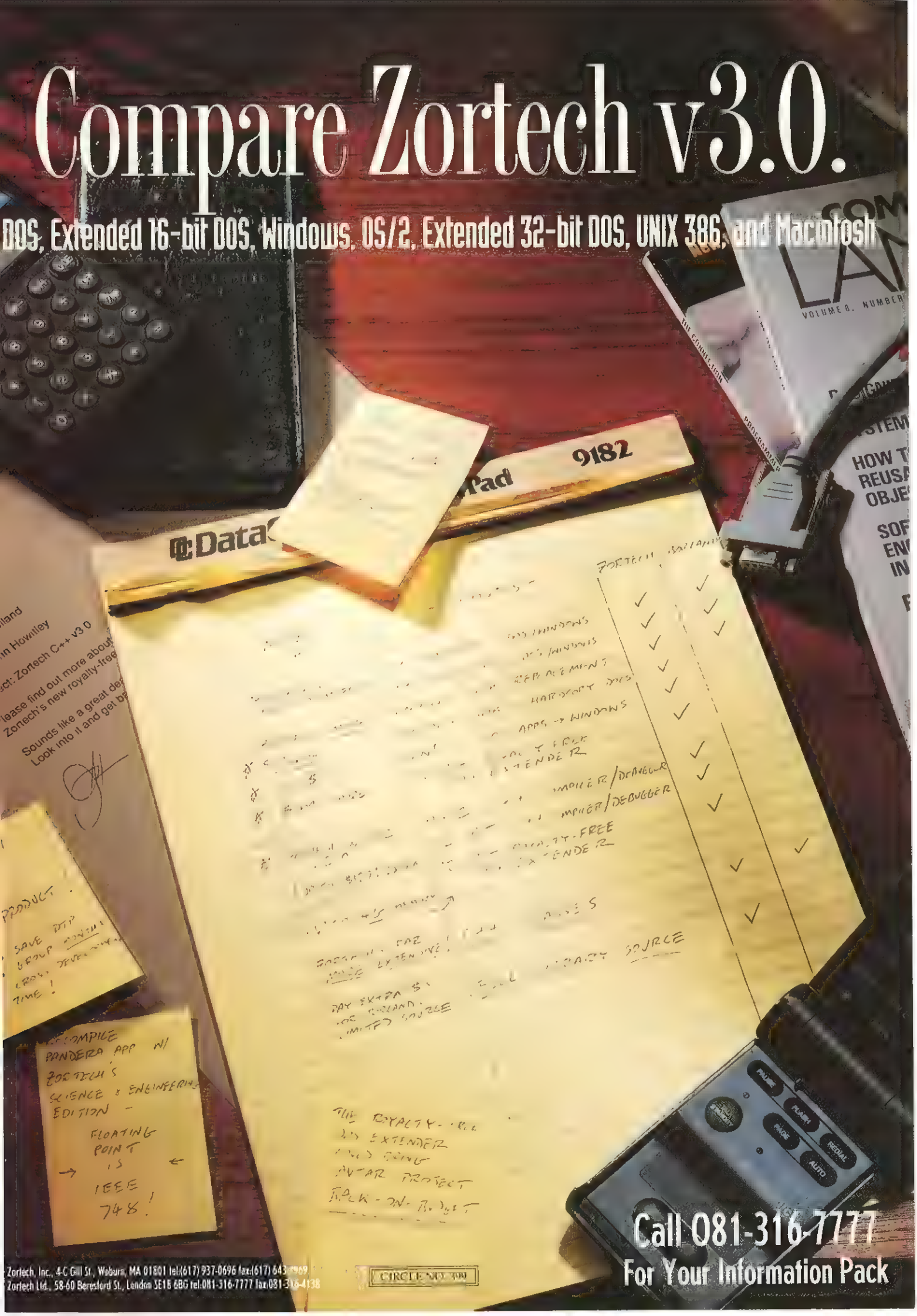
```
; now scan through the command tail
; for the first non-space character
@@sp_loop:
  cmp     byte ptr es:[di], ' '
  jne     @done_sp
  inc     di
  jmp     @sp_loop
@done_sp:
; di points to first non-space character
; display the signon message
  mov       ah,9
  mov     dx,offset signon ; signon message
  int       21h
; read the next (non-space) character
  mov     al, es:[di]
  cmp     al, '0'         ; int 60h?
  jne     @int_check1
  mov     al, 60h
  jmp     @set_int
@int_check1:
  cmp     al, '1'         ; int 61h?
  jne     @int_check2
  mov     al, 61h
  jmp     @set int
@int_check2:
  cmp     al, '2'         ; int 62h?
  jne     @int_check3
  mov     al, 62h
  jmp     @set_int
@int_check3:
  cmp     al, '3'         ; int 63h?
  jne     @int_check4
  mov     al, 63h
  jmp     @set_int
@int_check4:
  cmp     al, '4'         ; int 64h?
  jne     @int_check5
  mov     al, 64h
  jmp     @set_int
@int_check5:
  mov     ah, 9
  mov     dx, offset error_message
  int     21h
  ; return code is 1
  mov     ax, 4c01h
  int     21h

; now change the specified interrupt
; to the address of the new one
@@set_int:
; save interrupt number to replace
```

```
  push    ax
  mov     ah, 35h
  int     21h
  mov     ax, es
; is interrupt vector unprogammed?
  or      ax, bx
  jz      @set_new_int
; No.. display error message and finish
  mov     ah, 9
  mov     dx, offset error_message2
  int     21h
  mov     ax, 4c02h  ; return code is 2
  int     21h
@set_new_int:
; now program in the new interrupt
  mov     dx, offset new_int
  mov     ax, cs
; ds:dx has address of new routine
  mov     ds, ax
; retrieve new interrupt number
  pop     ax
  mov     ah, 25h
  int     21h     ; replace the interrupt

  mov     ah, 9  ; display 'loaded' message
  mov     dx, offset loaded
  int     21h

  mov     dx,((lastbyte-firstbyte)/16)+27
  mov     ax, 3100h  ; return code is zero
  int     21h        ; TSR
; ***** end of TSR loader section
main    endp
signon  db  'Sound playback system'
        db  ' for the PC',0dh, 0ah, '$'
error_message:
        db  'Unknown interrupt,'
        db  'valid types are:- ', 0dh, 0ah
        db  '0=60h, 1=61h, 2=62h, 3=63h,'
        db  ' 4=64h', 0dh, 0ah
        db  'Usage example: snd 0'
        db  0dh, 0ah, '$'
loaded  db  'has been installed'
        db  0dh, 0ah, '$'
error_message2:
        db  'Specified interrupt is already'
        db  ' in use, please try another'
        db  0dh, 0ah, '$'
cseg    ends
        end     firstbyte
```

Figure 3 - SND.ASM sound driver TSR program (Continued)

Compare Zortech v3.0.

DOS, Extended 16-bit DOS, Windows, OS/2, Extended 32-bit DOS, UNIX 386, and Macintosh



Call 081-316-7777
For Your Information Pack

Zortech, Inc., 4-C Gill St., Woburn, MA 01801 tel:(617) 937-0696 fax:(617) 643-7969
Zortech Ltd., 58-60 Berestord St., London SE18 6BG tel:081-316-7777 fax:081-316-4138

vector from NULL to the required address. If somebody else's code subsequently changes your vector and prevents access to your code, well stuff 'em!

The program in Figure 4 attaches itself to one of the free interrupts, which is specified on the command line. If the interrupt is already in use, it then returns a non-zero error level and doesn't go TSR. You can then try again with a different interrupt number.

The application passes the TSR a pointer (in DS:DX) to a table containing a list of note lengths, amplitudes and periods (the period being the inverse of the frequency, ie longer period, lower frequency). The program then changes the interrupt rate of the real time clock from its usual rate of approximately 18 Hz, to a more useful frequency of 10 KHz. This allows us to reproduce sounds of up to 5 KHz, which is about as much as you can expect from the little speaker inside your PC. The interrupt is then vectored to the sound playback routine, which scans through the table using the values contained therein, to generate the allegedly wonderful sounds which you have been waiting for.

NB: as the code stands, I haven't allowed the program to return to the caller until the sounds have finished. This is for various complex reasons, not least of which is that I have to leave something to write about another time!

Note( ) Well

The program shown in Figure 4 gives some simple examples of how to use the driver. I have declared a structure of type `notes`, which consists of a one byte value which is the period of the note (or the inverse of its frequency), its duration and its amplitude. The duration of the note is given in multiples of the 10 KHz interrupt rate, ie a duration of 10000 is one second, giving a maximum note length of 6.5535 seconds. The amplitude should be no more than about 100, otherwise you will notice nasty clicks in the sound.

To string together a number of notes, simply define an array of structures as I have done with the array `music[]`. You need not pass a list of simple notes to be reproduced as a series of boring bleeps. You can be much more creative. As anyone who has played with a music synthesiser will tell you, if you control the overall envelope of a sound, you can change its character entirely.

The function called `note()` performs such a task. This function requires parameters pertaining to the notes' frequency and envelope, in the widely accepted form of *Attack*, *Decay*, *Sustain* and *Release*, or ADSR for short. I have implemented this by defining the following:-

The attack time is the time which the note takes to rise from zero to full amplitude.

The release is the time to reduce the amplitude from full to half full level.

The sustain is the time to leave the level at half full.

```
/* Example use of Sound driver TSR */
/* For Small model Turbo C */

#include <stdlib.h>
#include <stdio.h>
#include <io.h>
#include <fcntl.h>
#include <dos.h>
#include <alloc.h>
#include <conio.h>

union REGS regs;
struct SREGS sregs;
struct notes
{
  unsigned char period;
  unsigned int duration;
  unsigned char amplitude;
} music[6000];

unsigned char tune[] = {60, 30, 80, 50, 70,
                        40, 60, 50, 0};
unsigned char tune2[] = {50, 128+10, 40,
                         128+10,
                        70, 128+10, 60,
                         128+10, 0};
main(int argc, char *argv[])
{
  char c, s[128], huge *bufbase;
  unsigned long length;
  int a, b, d, type, infile;
  FILE *fp;

  while (!kbhit())
  {
    for (b = 0; b < 3; ++b)
    {
      a = 0;
      while (tune[a])
      {
        note(tune[a++], 200, 500, 1000,
                                  1500);
      }
    }

    {
      b = random(6);
      for (a = 0; a < 3; ++a)
      {
        for (d = 0; d < 3; ++d)
          note(tune[b+a], 200, 500, 1000,
                                    1200);
        note(128+10, 200, 200, 800, 1200);
      }
      note(128+30, 10, 10, 10, 10);
    }
    a = 0;
    while (tune2[a])
    {
      note(tune2[a++], 5000, 2000, 2000,
                                    3000);
    }
    space(random(3)+4);
  }
  explode();
  for (a = 0; a < 10; ++a) gun();
  explode();
}

space(int n)
{
  int a, b;
  for (b = 0; b < n; ++b)
    for (a = 0; a < 100; ++a)
      beep(a, 100*b, 60);
}

note(int period, int attack, int decay,
                 int sustain, int release)
{
  struct notes m[300];
  int a, position=0;

  if (period & 128)
  {
    delay((period & 127) * 10);
    return;
  }

  for (a = 0; a < 100; ++a)
  {
    m[position].amplitude = a;
    m[position].duration = attack / 100 + 1;
    m[position++].period = period;
  }

  for (a = 0; a < 50; ++a)
  {
    m[position].amplitude = 100-a;
    m[position].duration = decay / 50 + 1;
    m[position++].period = period;
  }

  m[position].amplitude = 50;
  m[position].period = period;
  m[position++].duration = sustain;

  for (a = 0; a < 50; ++a)
  {
    m[position].amplitude = 50-a;
    m[position].duration = release / 50 + 1;
    m[position++].period = period;
  }
  m[position].duration = 0;
  play(m);
}

beep(int period, int duration, int
                                amplitude)
{
  struct notes m[2];
  m[0].period = period;
  m[0].duration = duration;
  m[0].amplitude = amplitude;
  m[1].duration = 0;
  play(m);
}

play(void *s)
{
  regs.x.dx = s;
  sregs.ds = _DS;
  regs.h.ah = 0;
  int86x(0x60, &regs, &regs, &sregs);
}

explode()
{
  struct notes m[6002];
  int a, amp;
  amp = 60;
  for (a = 0; a < 6000; ++a)
  {
    m[a].amplitude = amp;
    m[a].duration = random(5)+5;
    m[a].period = random(60)+1;
    if (((a % 100) == 0) && (amp>1))
      --amp;
  }
  m[a].duration = 0;
  play(m);
}

gun()
{
  struct notes m[602];
  int a, amp;
  amp = 100;
  for (a = 0; a < 100; ++a)
  {
    m[a].amplitude = amp;
    m[a].duration = random(5)+5;
    m[a].period = random(60)+1;
    if (((a % 20) == 0) && (amp>1))
      --amp;
  }

  m[a].duration = 0;
  play(m);
}
```

Figure 4 - Example app using sound driver

IN HARD TIMES, YOU CAN'T AFFORD TO GO SOFT.

In a tough market, everybody has to work harder. And that goes for your computer, too.

DR DOS 6.0 revolutionises the way in which data is stored on your hard disk, effectively doubling its capacity. (A hefty benefit, this – the cost and inconvenience of upgrading your hard disk doesn't bear thinking about.)

Feature Comparison.

	DOS 5.0	DR DOS 6.0
Full DOS application support	✓	✓
Up to 627kb memory on i386 & i486	–	✓
Up to 621kb memory on 286	–	✓
Up to 612kb memory on 8088/86	–	✓
Up to double hard disk capacity	–	✓
Read & write disk cache	–	✓
Disk optimiser	–	✓
Preloadable task switcher	–	✓
On-line hypertext user manual	–	✓
Password security system	–	✓
PC-to-PC file transfer utility	–	✓

Furthermore, the best disk caching on the market will speed up your computer's performance by up to five times. It's particularly effective with Windows, by the way.

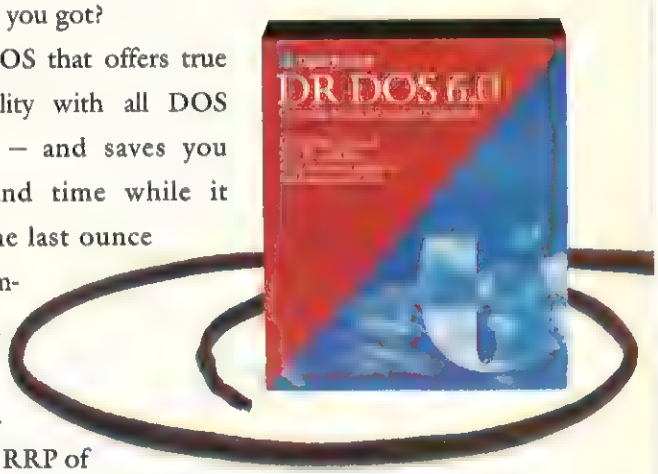
DR DOS 6.0 promotes better performance, too, by reorganising the data on your hard disk, so that it's more rapidly accessible.

Add the last word in memory managers, which frees up to 627Kb of RAM to run your software and which really comes into its own when a network is loaded. (Incidentally, we've extended this feature to all PCs, so that XT and 286 users can now reap the benefits of extra memory.)

Add a powerful task-switcher, which allows the user not only to switch between applications, but to cut and paste data between them.

Add a supremely-friendly graphical interface, a watertight security system and a useful file transfer facility, and what have you got?

A DOS that offers true compatibility with all DOS software – and saves you money and time while it wrings the last ounce of performance from your computer.



At a RRP of only £79, you should take a long, hard look at DR DOS 6.0. Frankly, you'd be soft not to.

DR DOS 6.0. A HARDER TASKMASTER.

To find out the full facts about DR DOS 6.0 attach your business card or complete the following:-

NAME: _____ ADDRESS: _____ DS6/EXE/1

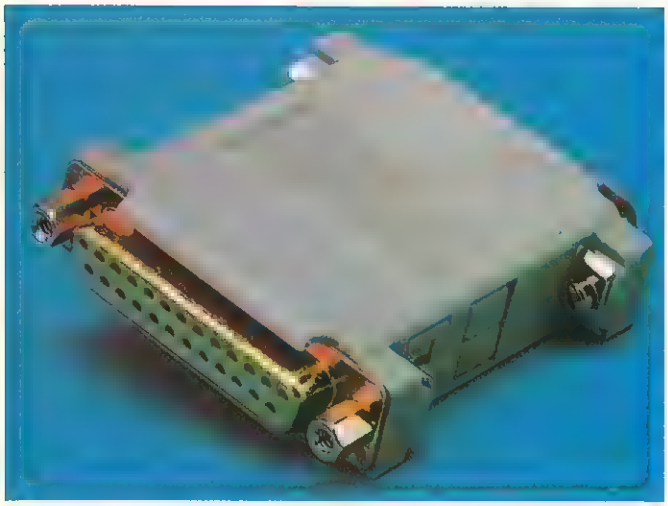
COMPANY: _____ TELEPHONE: _____

I am a business user ☐ dealer ☐ MIS ☐ other ☐

Please return to: Customer Service, FREEPOST, Digital Research (UK) Ltd, Oxford House, Oxford St., Newbury, Berks, RG13 1BR.
Tel: 0635 35304 Fax: 0635 35834

Digital Research
WE MAKE COMPUTERS WORK

CIRCLE NO. 400



How many users of your software paid for it???

SOFTLoK International Limited was established in 1987 with the introduction of our SOFTLoK and SOFTLoK PLUS devices to combat the ever increasing problem of software piracy. Our SOFTLoK range of software protection devices are used by hundreds of software developers from small consultancies to large multinationals. To cope with the ever changing needs of our customers we have developed SOFTLoK II which combines the programmable features of SOFTLoK PLUS with a low unit cost similar to our original SOFTLoK product.

SOFTLoK II™

The Next Generation

SOFTLoK II units are programmable devices containing read/write memory protected by a password. Both the memory and the password can be changed at any time using our routines in your application software. Easy to use menu-driven software is provided to allow small or large batches of SOFTLoK II units to be programmed with their initial data & passwords ready to be sent out with the protected software product.

As SOFTLoK II units plug into the parallel printer port they can be installed or removed in seconds.

Price: 1-19 £16.90, 20-49 £15.40, 50+ £13.70

Evaluation kit £20 (SOFTLoK II, manual & software)
All prices excl. VAT and delivery

- For IBM PC, PS/2 and compatibles
- Uses parallel printer port
- Totally transparent to printer
- Secure data & password can be changed from your application software
- Cascadeable
- 240 bytes of secure read/write memory
- 8 byte (64 bit) password
- No programming adaptors required
- Easy to use SOFTLoK II setup software
- Routines ready to link with various compilers
- Easy to follow manual

I-MEX House, 40 Princess Street,
Manchester, M1 6DE, England,
Tel: 061 228 7379 **Fax:** 061 236 6890

The decay is the time to return to zero amplitude.

In addition, I have defined any note with a period value which has the top bit set to be a gap or rest note. When the top bit is ANDed off, this leaves the amount of time (in milliseconds) to wait, without any sound issuing from the speaker.

If you look at some of the other functions defined, there names are self explanatory, eg `explode()` gives a fairly convincing explosion by generating a sequence of random frequency and duration notes of varying amplitude. The idea is that this is more or less an approximation of low frequency white noise, which when applied to the correct envelope shape, makes an explosion noise. I have exploited this further in the function `gun()` which imitates a machine gun.

You will (I hope) notice that all of the sound generator functions first define an array of structures of type and then proceed to fill it up in interesting and unusual ways. The sound playback is then performed by calling the function `play()`. This little routine simply loads the relevant 8086 registers up with a pointer to the data array which you have conveniently passed to it. The register pair in question are the `DS:DX` pair. I have set DS to the pseudo variable `_DS`, which in Turbo C (small mode) refers to the data segment where the variables are stored. If you work in a compiler other than Turbo C, please refer to your operator's manual, as the details are found to be different.

Is Anything There?

If you try to access a software interrupt that hasn't been defined or is null, the computer hangs. It therefore behoves you, as a responsible programmer, to check whether the driver has been installed or not. This is simplicity itself.

The first check to make is to ensure that the interrupt being called has a non-null vector, use `getvect()` or similar. Once you have determined that the interrupt is in use, read the bytes directly preceding the interrupt location. You will see the string of characters *SNDPLAYR*. This indicates that the interrupt under scrutiny is indeed the sound player.

Well, there you have it. PC sound in a nutshell. Who needs Sound Blasters and ADLIBs, eh?

EXE

Aidan Ruff is a practising programmer, hardware designer, financial genius (so he says). Notable hardware designs include co-designing the World's First Video Jukebox (quad processor, multi-tasking, 8085s!). A couple of industry design awards are hanging on the wall, including one from Arizona Microchip for making a washing machine micro controller talk! Aidan is the co-author/designer of SoftSpeak, reviewed elsewhere in this issue.

There are a record number of different ways of obtaining the code given in this article. It is available on CIX in the conference named softspeak/demo as EXESOURC.ZIP. Non Cixen modem owners may download it from Aidan's Olive Grove BBS (091 2280427 1200-9600 baud) between 6PM and 9AM, plus all weekend - log on as user name Exe Magazine with a password of EXE. The file EXESOURC.ZIP will appear in the files listing. (If you like the tone of the Olive Grove, please feel free to log in under your proper name.) You can ring Aidan on 091 2280513, and he will sell you a disk with all the items required for £5. Or you can send a disk and SAE to the .EXE offices, following the instructions on page 1. Please mark you envelopes 'SOUND'.

File locking

Lock up your data! Multi-process environments can easily foul up files, if the programmer does not discipline his programs. Peter Collinson explains how it's done.

Since the beginning, UNIX has been a multi-tasking system. This means that it's common to make programs worry about parallelism. For example, it's no good using a temporary file called `/tmp/tmpfile` when several instances of the program can be running together in the machine. Each instance will want to access the temporary file and will need that file to be unique to them.

On the other hand, there are many occasions where different processes will wish to read or write the same file simultaneously. The system needs to support this, so that there will be predictable results. Programmers need rules for what will happen in these circumstances.

You may think that simultaneous access will happen rarely. In fact, it's an everyday occurrence on a UNIX system. Let's think about the email system as a simple example. The email system will deliver your mail by placing it at the end of a file somewhere on disk. You will read mail by examining that file. What happens if you are reading mail at exactly the same time that a new bit of mail comes in? We are now in the situation where we have simultaneous access to the email file.

We want to make sure that the new mail is not lost. You may be deleting messages six and ten from your mailbox at the time new mail arrives. We don't want these messages suddenly to reappear. We need some way of sequencing the operations that are made by two independent processes so that the outcome is predictable.

On UNIX the two processes will co-operate and use a lock on the file. When a process wants to change the email file it will attempt to lock it. If the lock succeeds, then the process *owns* the file and can change it. After it has finished it will release the lock. If the process cannot lock the file, it will wait. Of course, we have to ensure that the programs all use the locks. This means that only one process is changing the email file at once, so things are predictable and we can program accordingly.

Do you need to lock?

Before you start worrying about using locks, you must ask the question: are locks really needed? Often you can code things so that explicit locks are not required.

Let's look at file I/O. When used with files, the `read` and `write` system calls are known to be atomic. If you use the `write` system call to put some data on a file, then that data will be written in one indivisible operation. If you like, you are guaranteed that all the bytes that you write will appear next to each other on the file.

Similarly, the `read` system call is also guaranteed to return data from the file such that it is coherent. Writes to the file will not be happening during a read.

If two processes have the same file open, then they will each have an individual pointer into the file. This pointer is manipulated by the `seek` system call and automatically moved on by `read` or `write`. If the file contains fixed length records, then one process can happily use `seek` and `read` to access the data while another uses `seek` and `write` to modify it.

However, things are more complicated when two or more processes are writing to the file. You will need to worry about locking if several writes can be made to the same position in the file at the same time.

Locking will also be needed if an operation on the file consists of more than one action. It depends on the application, but it's typical to read a record, change part of it and write it back. If someone gets in and changes the file *after* you have read it but before you can do the write, then the data will be incorrect. You must lock the file before the read and retain the lock until after the write.

However, you still may not need to lock every access to the file. A process that is reading the data is guaranteed to obtain a complete record when output processes use the `write` system call. This ensures that that writes are atomic. By careful coding, it's possible to require that only writer processes lock the file.

Log files

Maintaining log files is another common application that involves several processes writing to a file. You might try to code this like:

```
    fd = open(log, O_WRONLY);
/*  seek to end of file */
    lseek(fd, 0L, SEEK_END);
    write(fd, msg, msglen);
    close(fd);
```

Of course, you *must* test for success or failure of these operations. Apart from that the code may look sound to you. We are doing an atomic write call because we know about that.

This code sequence may often seem to work, because of the way that UNIX schedules things. However, it won't work all the time. Consider two processes executing the code in parallel. It's possible for both of them to enter the `write` call at exactly the same time. Both will have file pointers positioned at exactly the same place in the file. The kernel will do an internal lock on the file to guarantee atomicity and one process will be executed first to write its message. The second will then simply overwrite the first message, leaving unpredictable things in the file.

We could deal with this by locking. But writing log files is common and UNIX supplies a special mechanism to help. You can open a file passing a special flag:

Which applications generator offers all of the following features?

- No Runtime or Royalty Payments for any generated C Code
- Fully integrated Multi-Version Dictionary and Database
- WYSIWYG Screen, Report and Menu generation
- Data Managers

  PRO-TREE
  Btrieve
  Oracle
  Paradox
  C-TREE
  C-ISAM/INFORMIX
  dBase III

- Operating Systems

  DOS - Single or Multi-User
  UNIX
  XENIX
  QNX
  AIX
  Ultrix
  More environments on the way!

- Style of C Code

  K & R
  ANSI
  Proto

- Support/training

  East Anglian base
  On-site training available
  Modem support available

PRO-C

The Open Approach.....

For further details, contact ITEL on 0206 262244/0902 763786 or write to:

ITEL
PO Box 1229
Colchester
Essex CO3 3LN

ITEL
4 Wychbury Road
Merry Hill
Wolverhampton WV3 8DN

```
fd = open(log,
        O_WRONLY|O_APPEND);
```

The O_APPEND tells the system that the file is to be opened in append mode. What happens now is that file pointer is automatically set to the end of the file in each write system call. The write call becomes an atomic 'seek to end of file and write this'. Our logging routine can be:

```
fd = open(log,
    O_WRONLY|O_APPEND|O_CREAT,
        0644);
write(fd, msg, msglen);
close(fd);
```

I have added the O_CREAT flag to force the file to be created if it doesn't exist. Again, we have managed to do without locking. Many processes can now write to the log file, and and use the write system call to ensure that their data is written intact.

Simple locks

Once you have decided that you do need to lock a file or a portion of a file to guarantee exclusivity, then how do you do it? Unfortunately, early UNIX systems did not support file or record locking and a variety of different methods have grown up over time. These days UNIX may provide one of several different types of locking mechanism in the kernel. Worse, it may be that none of the system call methods will be available on your system.

If you have no mechanism built into your system you can create simple locks on files by using the standard system calls. Many extant program suites for UNIX still use this type of approach.

To make a lock, we will create a file. The lock is on when the file exists, and off when the file is not present. The original idea was to use the creat system call making a file that is not writeable:

```
fd = creat(lockfile, 0);
```

A second creat call will fail because the file already exists and is not writeable. To remove the lock file, we simply delete the file.

This works for everybody except superuser. A second creat call will not fail for superuser because the file access permissions are ignored. This is a kind of historical 'feature'. We need a better way because we cannot guarantee that the superuser will not start the program.

We cannot do a two stage operation like stat the file and call creat if it exists because there is a window between the two system calls where the state of things might alter. We need a single system call that will create a file. One candidate is the link system call.

The link system call creates a directory entry that points to an existing file. It doesn't create a new file, but makes a new name for a file that already exists on disk. After the link system call, two names are present in the file system that point at the same file contents. We say that the file has two *links*. The new link can be on the same directory, or in a completely different part of the file system tree. If it is created in the same directory, then it must be a unique name.

The link system call has the general form:

```
rv = link("old", "new");
```

This will make a link called new that points at the contents of the file old. The routine returns success or failure status.

If you delete one of these links, then the contents of the file will not disappear from the disk. The old file does not *own* the contents, rather you should think that the contents belong equally to all the links that point at them.

The system stores the total number of links that point to a particular file contents. It will only free the disk space occupied by the contents when the last link is deleted. The system call that deletes files is called unlink because it is really only deleting links, not removing names from directories. When the link count for some contents drops to zero, then that bit of the disk is freed. We can use all this to make a simple rename or mv program:

```
rv = link("old", "new");
unlink("old");
```

This first creates a link to the file old called new and then deletes old leaving new in the file system. This is a cheap way of moving a file around a file system; moving the name and not the contents. For technical reasons, it will only work within one file system and the real mv program is a little more complicated.

We are going to make use of the property that the link call will fail if the file new exists. This will be returned as an error status by the system call. In general, none of the other reasons for failure will be triggered by our routine. The link call will behave like an atomic *test-and-set* for file names. We have an indivisible operation that will create a known file. Here we go:

```
lockit(lock)
    char *lock;
{
    int fd;
    char t[256];

    /* make a temporary file */
    /* in the same directory */
    /* as lockname */
    (void) sprintf(t, "%s.%d",
            lock, getpid());
    fd = creat(t, 0444);
    if (fd < 0)
        return SHOULDNTHAPPEN;
    close(fd);

    /* Now use that file */
    /* as a base for linking */
    /* can we create lock? */
    if (link(t, lock) < 0)
    {
    /* No, file exists */
        unlink(t);
        return EXISTS;
    }

    /* remove tmp file */
    /* leaving lock */
    /* in place */
    unlink(t);
    return LOCKMADE;
}

/* remove a lock */
unlock(lock)
    char *lock;
{
    unlink(lock);
}
```

The routine first creates a temporary file with some unique name. It should be in the same directory as the lock file, but we don't really care what it is called. It's just an object to link to. I have used the lock file stem and added the process id. We will expect the file creation to succeed; it should only fail during the debugging.

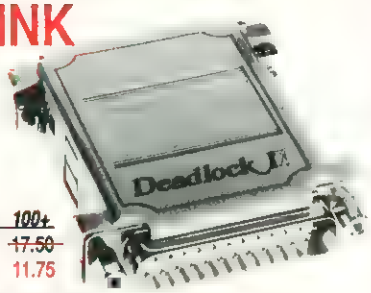
The link system call is used to make the lock file. It creates the lock file as a link to the temporary file that we have just made. If the link succeeds, we delete the temporary file leaving the lock file in place.

The link call will fail if the lock file exists already. We return this fact to the caller. The caller needs to implement some code to enforce a delay before the routine is retried. Simply calling lockit again is decidedly antisocial.

Once this routine has created the lock, the file will stay in the file system until deleted. We have to worry about this a little. Programs should catch all possible signals and make sure that the lock file is removed when an exception occurs. Even then, a program may crash leaving the lock file in place.

Unfortunately, this is really a 'feature'. There are some schemes that use heuristic methods of guessing that the lock owner *might* have died. But in general, you cannot get around this problem easily. A lock should really be a kernel object that is released when the process dies.

DON'T THINK DONGLE, THINK



PRICES	5-50	51-100	100+
	~~16.50~~	~~13.75~~	~~17.50~~
	13.75	12.50	11.75

THE KEY TO THE FUTURE

- ★ Ideal for Demos, Rental etc
- ★ Read/Write facility
- ★ 16 Bytes Password
  10 bytes memory
- ★ Programmable by software developer
- ★ Routines supplied for most common compilers
- ★ Supports DOS, Windows
- ★ Immediate delivery
- ★ Guaranteed for 100,000 writes to key during run time
- ★ FREE software for protection of .EXE or .COM programs
  To be used as demo, rental or full protection
- ★ Attaches to paralell port of AT, 336, 486 compatibles
- ★ FREE evaluation on trial & return basis for period of six weeks. Please call for further conditions.
- ★ Do you wish to save time, money & your software?

Call us now for a special introductory Christmas offer

Call us now on: TEL: 081-343 0734 FAX: 081-346 2672
BL Computer Security, 101 Hendon Lane, Finchley, London N3 3SH

All Trade Marks acknowledged

CIRCLE NO. 404

Configuration Management

PVCS and PolyMake for UNIX platforms...

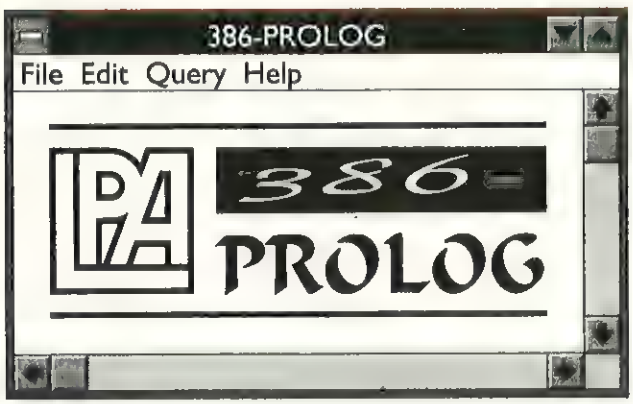
- DEC
- Bull
- DataGeneral
- ICL
- Motorola
- NCR
- Nixdorf
- Prime
- Siemens
- Unisys
- and others...

Readmar Systems
L I M I T E D

Tel (+44) 071 625 5255
Fax (+44) 071 624 9404

CIRCLE NO. 405

Windows 3!



Yes, it's here! LPA 386-PROLOG is now available in a version for Windows! Just like the DOS-extender version, it is a genuine 32-bit Prolog compiler which can directly access up to 4G (4096M) of memory. Only this time, it is fully integrated with the world's most popular GUI: Windows 3!

Logic Programming Associates Ltd
Studio 4, Royal Victoria Patriotic Building
Trinity Road, London, SW18 3SX, England
Tel: 081 871 2016 - Fax: 081 874 0449

CIRCLE NO. 406

F77L-EM/32 & Lahey Ergo OS/386

Port mainframe programs as large as 96MB to 386/486's with this 32-bit DOS-Extender compiler. The winner of PC Magazine's 1988 Technical Excellence Award just got better. New Version 4.0 includes: Programming Tools, Popular Fortran 90 features, Virtual Memory Support, DESQView support, New Documentation and Free Unlimited Runtime Licenses. F77L-EM/32 and OS/386. *£875.00 plus VAT*

F77L

The fastest real-mode compiler available. F77L can take advantage of your 386 PC by generating 32-bit instructions. New Version 5.0 includes: Fortran 90 features, Weitek support, and Video Graphics. *£375.00 plus VAT*

Lahey Personal Fortran 77

Version 3.0: Full ANSI 77, Editor, Debugger, Linker, Library Manager, Microsoft and Borland C interfaces. A great learning tool at an unbeatable price. *£79.00 plus VAT.*

System Science

3-5 Cynthia Street, London N1 9JF
Tel: (071) 833 1022 Fax: (081) 837 6411

CIRCLE NO. 407

System calls for locking

There have been several attempts at putting locks into the kernel. You may find one or more of these calls on your system. The `flock` system call was added into BSD systems to perform file locking. It simply permits locking of whole files.

Commercial variants of UNIX added a routine called `lockf`. It became popular and found its way into the early UNIX standards created by /usr/group. This in turn influenced AT&T, and the routine appeared in System V.

AT&T had also created a more general mechanism using the file control system call `fcntl`. It is possible to implement both the `flock` and `lockf` calls using this general mechanism. POSIX has adopted the `fcntl` method, so hopefully things will begin to get more portable.

I am going to look at the `lockf` routine, largely because it's easy to use and gives you the flavour of what is happening. It has the general form:

```
ret = lockf(fd, cmd, size);
```

The `fd` is a normal open file descriptor that is the file that is to be locked. The file must have been opened as writeable. Locks are stored in a central table in the kernel. The table holds an internal id of the file, the owner and details of the region for each region of a file that is locked. The internal id for the file is derived from the file descriptor. An entry in the table is made when locks are created, and deleted when the region is unlocked.

The cmd parameter specifies the action to be taken. There is a total of four commands. `F_ULOCK` unlocks a previously locked section of the file removing the entry from the table. The system deals correctly with the situation when you want to a whole file and only unlock a portion of it.

`F_LOCK` locks a section of the file for exclusive use; this creates a table entry. `F_TLOCK` tests for a lock in a section and locks it if there is none. Finally, `F_TEST` simply tests a section for a lock.

The `size` parameter represents the number of bytes to be locked or unlocked. The current position in the file, maintained by the seek pointer, is used as the base of region in the file. If `size` is positive, the region stretches forward in the file from the current position. If negative, it stretches backwards in the file. This allows us to lock portions of a file.

If `size` is zero, the section from the current position to the largest file offset is locked. This locks the whole file. Our simple `lockit` routine becomes:

```
static int ld;

lockit(lock)
   char *lock;
{
   int rv;

   /* open the lock file */
   ld = open(lock,
          O_WRONLY|O_CREAT,
          0444);
   /* assume open worked */
   /* this is dangerous  */

   rv = lockf(ld, F_LOCK, 0L);
   return rv;
}

unlock(lock)
    char *lock;
{
    (void) lockf(ld,
                 F_ULOCK, 0L);
    /* should check for */
    /* success or failure */
    close(ld);
}
```

There are several benefits over our previous version. We don't have to worry about locks existing after the process has died. When the process dies, the lock dies with it.

Also we don't need any sleep code to recall `lockit` if the lock is not created. The program will wait in the `lockf` call until the lock succeeds. It will wait forever, so we might need some code to stop that happening. Alternatively, we could code things to use `F_TEST` and `F_TLOCK`.

This example is very simple and I haven't worried about locking portions of files. There also are other issues involved with locking. For instance, the lock file mechanism above is only advisory. You can ignore the lock file and take the action that the lock file is protecting against. Advisory locking demands that all processes that access the file agree to obey the locking mechanisms and implement the needed functionality.

We might like to lock a file against access by naive processes, making the locking 'mandatory'. This can be done with `lockf` by setting the file permission specially. I think that this is just a dreadful way of going about things.

Deadlocks

If a program or a suite of programs uses more than one lock then you must be careful or deadlocks can easily arise. A deadlock or 'deadly embrace' means that two (or more) processes are stopping each other from running. Let's say that one program does `lock(a)` and second program does `lock(b)`. Here `a` and `b` are files or portions of files.

If the first program now tries to `lock(b)` it will sleep waiting for the second program to release the lock. Meanwhile, the second program does a `lock(a)` waiting for the first program to release the lock. Both programs are now sleeping, waiting for each other.

It's easy to do this unless you are careful. One rule to follow is that locks must always be applied and removed in the same order in every program that uses them. Locks should always be removed in reverse order of application, so

```
lock(a);
lock(b);
...
unlock(b);
unlock(a);
```

If you cannot arrange ordered locking, then you can use the built-in deadlock detection that the `lockf` routine supplies. If a deadlock between two processes is happening, then the `lockf` routine will return an error. The routine returns -1 and loads the error EDEADLCK in the external variable `errno`.

This works because the locks are stored internally in a table containing a list of regions in files. It's possible to compare regions to detect deadlocks. The deadlock detection only works between two processes. It's certainly possible to envisage a scenario where many processes are dealing with several different locks and getting into deadlock situations. Program with care.

More reading

I decided not to explain the `fcntl` system call for locking because its fairly complicated and the article would have grown too much. I imagine that most systems will continue to support the simpler `lockf` call since it can easily be implemented on `fcntl`. You must consult the manual pages that come with your system to find the locking system that is available to you.

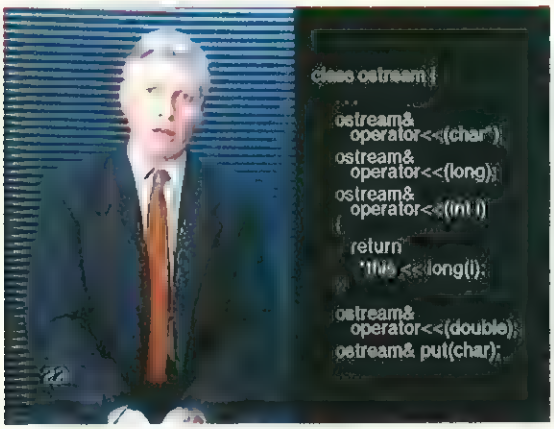
For a longer description of locking see: *UNIX System Programming* by Keith Haviland and Ben Salama published by Addison Wesley.

EXE

Peter Collinson is a freelance consultant specialising in UNIX. He can be reached electronically as `pc@hillside.co.uk` *(although your mailer might be happier to put the address the other way round) or by phone on 0227 761824.*

VIDEO COURSE

Learn C++



NO RUSH!
NO TRAVEL!
NO HOTELS!
ONLY £299.95

Learn C++ Now!
The great exodus of programmers from C to C++ has begun! Since C++ builds on C, it's the easiest OOP language to learn. That's why it's called "the language of the 90's".

Why the rush?
PRODUCTIVITY! Yes, C++ programmers can write programs in less time requiring less maintenance. Large projects become much easier to manage.

Unfortunately, learning C++ can be very costly. Classroom instruction is expensive even without the travel and hotel costs. Of course, not learning will cost you even more in the long run. Now there is an alternative!

The top C++ video tutorial at the lowest possible price.
The C++ video tutorial from Zortech is the ultimate C++ training tool for work or home at only £299.95. It comes on six VHS video tapes containing 32 clear, extensive tutorials.

Used in conjunction with the concise workbook and tutorial disk, you will find everything you need for fast-track C++ tuition.

The course is generic (i.e. compiler and hardware independent) and is available with or without the award winning Zortech C++ Compiler for MS-DOS and OS/2.

Start writing C++ code within a week.
As a C programmer, you will start producing C++ code within a week of concentrated use of this course. Alternatively, spend just an hour a day watching the video and working through the suggested exercises to learn C++ in only six weeks!

The Leader in video tutorials.
If you don't already know C, you can join tens of thousands of programmers world-wide who have learnt C with the Zortech Complete C Video Course, described as:

"An excellent bargain ...
... I heartily recommend"
Gary Ray, PC WEEK

Save your company thousands of £££'s.
One programmer can train for only £299.95, but you can train ten programmers for just:

C++ Course	£299.95
9 Extra Workbooks	£179.55
Total 10 students	£479.50

Yes! Only £47.95 each! (With all the FREE refresher courses you need!)

Now, Zortech with its new C++ Video Tutorial has refined the art of video tuition and presentation even further for the 1990's.

Only £299.95 complete
- *Six Videos with 32 lessons*
- *256 page workbook*
- *Tutorial disk*
- *Compiler & hardware independent*
- *NTSC or PAL format*
- *Tax deductible*

Don't delay, order now!
Just mail the coupon or call the order hotline for same day shipment.

USA: Zortech Inc.,
1165 Massachusetts Avenue,
ARLINGTON, MA02174
Voice: 617-646-6703
Fax: 617-643-7969

EUROPE: Zortech Ltd.,
106-108 Powis Street, LONDON
SE18 6LU
Voice: 44- 1-316-7777
Fax: 44- 1-316-4138

IN EUROPE CALL:

44- 1-316-7777

ORDER FORM
Please rush me these items:

Qty	Description	Price
___	C++ Video Tutorial	£299.95
___	Extra C++ Workbooks	£. 19.95
___	Zortech C++ Compiler	£129.95
___	Developer's Edition	£299.95

Please add £5 + VAT postage in UK
All UK orders please add 15% VAT
International shipping charged at cost

Name _____
Company _____
Address _____
_____
_____
Phone _____
MC,VISA or CHQ_____ Expiry ____
Card No. _____ .EXE 12/91

Menu-IX | System-IX | DBMS-IX | Sysix | AI | Spell | Merge | Lock | GURU | Crypt | Tawk | Tgrep

Cut | More | Term Ctl | Where | Sort | Grep | Idle | CB Trees | Evaluate | Awk | Banner | C Source | Fgrep | Help | Head | Tail

Xenix | Xenix Net | Unix | Interactive | System V | TCP/IP | DOS | Token-Ring | OS/2 | Ethernet | Novell Lan | Mic Net | Aix | VP-IX | VMS | QMS

Xvale/Lnx, Store 'D'
55, Bedford Court Mansions,
Bedford Avenue, London.
WC1B 3AD U.K.

TEL: 071-636-8210 / 071-631-4818
FAX: 071-255-1038 24 Hours

Barclaycard Access

UNIX

Now! you can get into UNIX and VPIX Multi-user DOS) at affordable prices, with a leading specialist of the UK.
System V R 4: 8 console usr; BASE system, FACE, NFS,TCP/IP, RFS,DFS RPC,XDR,X11,Xt,Xview,Motif,(compat:BSD,XENIX),(Standards:SVID, POSIX,FIPS,XPG3),SDS,SCDE,DLL,ELF,FMLI,386/486/EISAsupported.

Interactive Unix SysV, REL 3.2
(Ver 2.21 Below) (New Ver 3.0 call for price)

S.V.R.4
£2395.00

Workstation Developer: 1-2 usr £1295.00 U/L usr £1995.00
Operating System, Vpix (DOS under UNIX),
Software developement system plus New 'C', CO-edit, Codewatch, Ten-Plus, Looking glass, XII windows runtime and development system, NFS/TCP/IP, and complete documentation.

Workstation Platform: 1-2 usr £855.00 U/L usr £1495.00
As workstation developer, excluding software development.

Network Developer: 1-2 usr £1155.00 U/L usr £1815.00
As workstation developer, excluding XII Windows.

Network Platform: 1-2 usr £695.00 U/L usr £1350.00
As Network Developer, excluding development items.

Application Developer: 1-2 usr £1095.00 U/L usr £1750.00
As Network Developer excluding network items.

Application Platform: 1-2 usr £495.00 U/L usr £1095.00
As Application Developer, excluding development items.

PLEASE NOTE!. All prices advertised , are prepared about 2 months in advance. Please CALL to confirm LATEST PRICES!.

UNIX ITEMS	
Motif Dev	£345.00
PC-DOS bridge	£175.00
Norton for unix	£195.00
Easy windows	£595.00
Ported Netware	£3995.00
SMB/ix	£895.00
Informix/Oracle	Call
Lotus 123	£495.00
LPI Cobol	£1095.00
LPI New 'C'	£595.00
LPI Fortran	£ 895.00
LPI Pascal	£895.00
LPI Basic	£595.00
LPI /PL/1	£1795.00
SVR4 Starter	£1295.00
SVR4Grph/De	£1795.00
MENUIX	£49.95
SYSIX -pro	£145.00
LPI (unix) C++	CALL
XSIGHT	CALL

SCO & Others	
Xenix Items	Call
Unix o/sys	£525.00
Unix s/dev	£613.00
Unix vpix	£321.00
Unix tcp/ip	£227.00
Unix nfs	£340.00
Open dsktp	£795.00
O/dtp s/dev	£1095.00
O/dtp servr	£1095.00
Guru reprter	£49.95
Syseditor	£49.95
Cshell	£49.95
Menuix	£49.95
Sysix Pro	£145.00
SystemIX	£295.00
PC-Interface	£170.00
Pc-int/wndw	£225.00
TCP/IP dos	£185.00
PC-Xview	£245.00
Terminl ctl	£695.00
PC-Xsight	£395.00

LNX

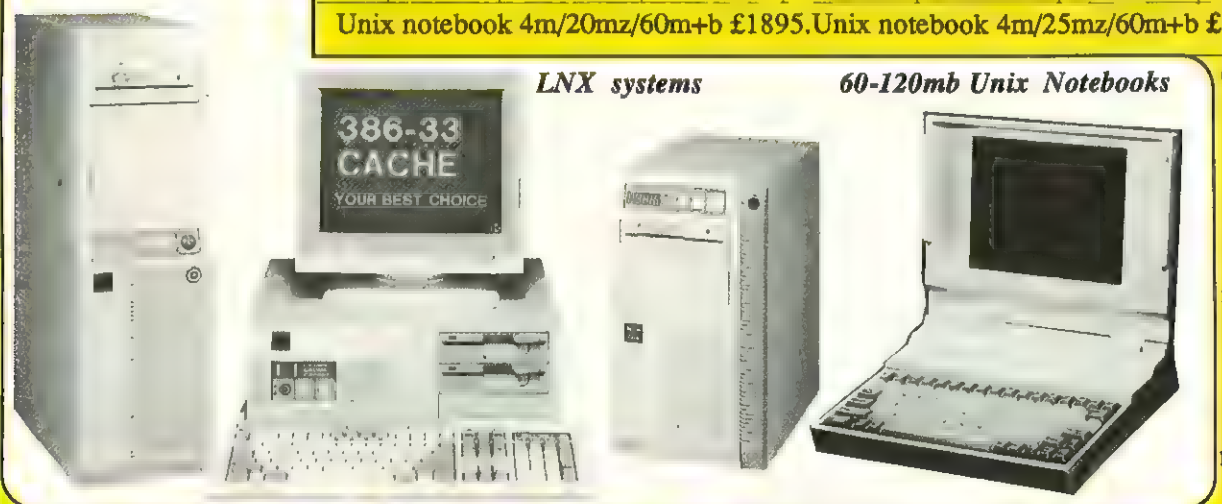
Base 1 level: Mini-tower, Mono screen & adapter with parallel port, Dual hd/fd ctrlr with 2S/1P/1G port, 102 keyboard, 1 Mb ram, 1 * floppy drive (5-1/4 or 3-1/2), Mouse.
Base 2 level: as base 1 + SVGA colour scrn & 512k svga card, 4 Mb ram, 5-1/4 &3-1/2 FD
Base 3 level: as base 2 + 1 Mb ram on SVGA card, 8 Mb ram on m'board.
Base 4 level: as base 3 + 16 Mb ram on m'board. Full tower case (e & oe)

Free Software Bundles

a)Sysix-pro/family,Cshell £295
b)Unix v2.02,Vpix,Sysix £395
c)Unix v2.02 Platform £475
d)Unix v2.02 Developer £695
These offers may vary. Some have Unix wk/dev 2.21 & SVR4 CALL! for up to date bundles

	386-16	386-25	386-33	386-40	486-33	486-50	
0mb	£567	£695	£880	£897	£1235	£2060	Base 1
40m	£792a	£920a	£1106a	£1122a	£1460a	£2285a	Mono
100m	£1250a	£1377a	£1563a	£1580a	£2105b	£2930b	Base 2
200m	£1445a	£1572a	**£1758b**	£1775b	**£2299b**	£3125b	SVGA
350m	£1925a	£2052a	£2238b	£2555b	£2780c	£3605c	Colour
600m	£2225b	£2352b	**£2538b**	£2555c	**£3080c**	£3905c	System

Unix notebook 4m/20mz/60m+b £1895.Unix notebook 4m/25mz/60m+b £1850



LNX systems *60-120mb Unix Notebooks*

Option A
150 MB
TAPE
£795

Option B
8 Port i/o
with
2 * Wyse
Terminal
£1395

Option C
1.2 Gigbt
h/d & ctrl
£2495

Compaq | Toshiba | IBM | Epson | Qume | NEC | Seagate | Wyse | Relysis | Zenith | Priam | Ranger

Books

Two Approaches to Better Programming

More about Eiffel

Best to acknowledge my disqualifications before getting too deep into Betrand Meyer's *Eiffel: The Language*. I am not, and never have been, an Eiffel user. However, in my defence, I know that I am far from alone in admiring the Eiffel language from afar, as it were. The point about Eiffel is that the previous book in which it starred, Meyer's *Object-oriented Software Construction*, contained the most lucid and convincing arguments for OOP that I have read. Where lesser authors attempting to explain the non-trivial concept of multiple inheritance get hopelessly bogged down (writing of, for example, class `boatplane` which inherits from classes `boat` and `plane` - not a situation I am faced with very often in *my* programs), Meyer leaves you thinking 'what a neat trick that is', and longing to try it out. Which I couldn't, because at that time, Eiffel was not available for MS-DOS.

Eiffel: The Language is published to coincide with V3.0 of the design (defined in the book), plus a general Eiffel *glasnost*: control of the specification of the language has passed from Meyer's company to a non-profit making consortium, there are now several implementations - including an MS-DOS version - and so on. Presumably the hope is to move Eiffel out of the college campus and privileged R&D section of the livelier corporates and 'onto the streets'. Given this hope, *Eiffel: The Language* must be a more important work than your run-of-the-mill language textbook.

The book disregards the K&R model of separating a tutorial/user guide from a formal reference manual; instead we have a single combined narrative, the various components flagged by italic notes in one margin and by 'road signs', in the other. The intention, says the author, is to avoid forcing the reader to hunt through both user manual and reference section of the book. He may have succeeded in doing this - it's hard to tell without having used the text 'in anger' - but at the cost of readability. As an essentially casual reader, I quickly started skipping the paragraphs (appropriately) marked with no-entry signs containing validity rules ('...If a Parent part for B in C contains an Undefined subclause, that clause is valid if and only if, for every Feature_identifier *fname* that it lists...'). I don't instantly wish to know about a language's validity rules; I want to know how to print 'Hello World', and organise files, and declare arrays. I won't be writing the compiler until next week, thanks. Notwithstanding road signs, the author's decision to mix reference and user material forces the reader to do extra, pointless work.

The chapter structure of the book is also innovatory. Meyer describes his approach as 'top-down'; one chapter describes how systems are built from classes, the next begins to supply the constituents of classes. The first mention of Eiffel's simple loop structure (P129) comes long after we have dealt with the problems caused by repeated multiple inheritance. (But there again, this may because loops are slightly frowned upon in Eiffel - you should be using a library iterator.) This makes for a lot of what my assembler calls 'undefined forward references' - a very generous set of cross-references supplied in the margins makes it acceptably light work to follow them up, but it wasn't *me* who brought up this business of not having to prop open a book in two places at once. It also means, as Meyer says, that the reader is required to take many things 'on trust'. This book would definitely be easier to read a second time; or perhaps I'm cursed with a bottom-up mind.

The book is organised as five main sections: an introduction, *The structure*, *The contents*, *Elements from Basic Libraries* and *Appendices*. These latter include syntax diagrams, an essay on language design (Meyer writing at his approachable best), and two chapters on the differences between Eiffel 3 and earlier versions. Apart from the road signs, the text is liberally illustrated with diagrams representing class hierarchies, data structures etc. It goes without saying that, as far as I (the non-Eiffel programmer) can tell, the text is completely thorough and scholarly.

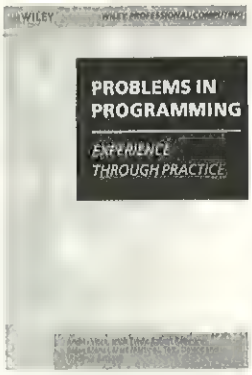
In some ways (and acknowledging that supporters of both camps will probably vigorously deny this) the Eiffel vs C++ debate is the natural follow-on to the old Pascal vs C campaign, now lost by the Wirth camp. On the one hand we have the deeply pragmatic Stroustrup, with his 'give the programmer enough rope' low-level, compatible-with-C approach. On the other, there is Meyer the Purist; who builds a completely new language from scratch so that he can eliminate all impurities (type casts are described in this book as 'sordid back-alley deals'), and who favours programming style over efficiency (Eiffel implements garbage collection because looking after heap memory is a job 'too important for the programmer'). I certainly don't advocate dreary and petty 'my programming language is better than yours' debates such as are frequently carried out in electronic conferencing systems. However, I do note that the flow of ideas thus far has been from Eiffel to C++; the former had multiple inheritance, generic classes and exception handling all implemented while they were still but a gleam in AT&T's eye. Eiffel offers plenty of other concepts not included in C++ (or not possible to include). The C++ programmer who disregards Eiffel is being a fool to himself. *WRW.*

Title: Eiffel: The Language — *Pages: 300*
Author: Betrand Meyer — *Publisher: Prentice Hall*
Price: £22.95 — *ISBN: 0-13-247925-7*

Books Received This Month

Advanced C++ Programming Styles and Idioms by J Coplien	Addison Wesely	£27.85	ISBN.0-201-548-550	pp520
Turbo Pascal for Windows 3.0 Programming by T Swan	Bantam	£26.99	ISBN:0-553-35293-8	pp876
Object-Oriented Programming in Turbo Pascal by K Weiskamp, L Heiny and B Flamig	John Wiley	£22.95	ISBN:0-471-52466-2	pp409

Good Techniques



Your boss has asked you to write a little piece of code which will extract a handful of records from his database. Ten minutes later you boldly stroll into his office and hand over a disk containing some Pascal code(!) that does the job in question admirably. Several hours later, he's hopping menacingly around the office, demanding your resignation. How were you supposed to know that he wanted all the 'Smiths' in the 'Yellow Pages'?

The moral of this story is that although a program will undoubtedly run as you intended ('cos you're a darn good programmer), it probably won't cope so well with unforeseen restrictions. For instance, a program for sorting 10 records will be fundamentally different to one that needs to sort 10,000.

Problems In Programming is not about sorting. Instead, the authors have provided the reader with over a hundred problems and their Pascal solutions. These are divided into eight topic areas which include coverage of recursive functions, reordering, graphs, controlling real-time processes and computer graphics.

The problems were actually taken from a series of competitions between 1977 and 1987 in Slovenia. The authors have devised an ingenious notation for categorising each problem and its associated solution.

The ID of each problem/solution pair contains enough information to determine the year in which the original problem was set, its level of difficulty and, of course, its topic area. Problems are kept separate to their solutions and this means that it is almost impossible to read the book from cover to cover (unless you're a journalist). You end up having to flick back and forth between problem and solution, imitating an organically grown equivalent to hypertext.

The problems are written in that rather bland style that you associate with an A-level Maths text book. In fact, several of the solutions require quite a substantial knowledge of mathematics, ranging from 'arithmetic progressions' to 'proofs by induction'. Luckily the solutions outline the problem-solving process step by step, revealing how a complicated problem can be broken down into a simple mathematical formula. Some of the logical problems are guaranteed to bring back fond memories of De Morgan's theorem, and the more theoretical ones should keep you amused 'till the early hours of the morning.

Problems In Programming provides you with many techniques that you could apply when programming a real application. Although the problems are not always realistic, there is a lot of good code to digest. The authors have shown how to produce optimum solutions if you've got the time. But if you only have ten minutes...

It's almost time to go home, but he's waiting for you. He tells you that he needs some more stuff from the database and adds that it won't take you a minute to put it all together. You feel like asking him how's his father? But you value your job...

Title: Problems In Programming *Pages: 327*
Authors: A Vitek, I Tvrdy, R Reinhardt, B Mohar, M Martinec, T Dolenc and V Batagelj *Publisher: John Wiley*
Price: £14.95 *ISBN: 0-471-93017-2*

Modus Vivendi

Our dedicated crossword setter, Eric Deeson, tries out a specialist package.

Unsolicited review software is often received with a groan by magazines. Not so in this case: as someone who has compiled crosswords for over 20 years, and still uses little technology apart from a word processor to help, I was most impressed from the start with the potential of *Crossword Modus*.

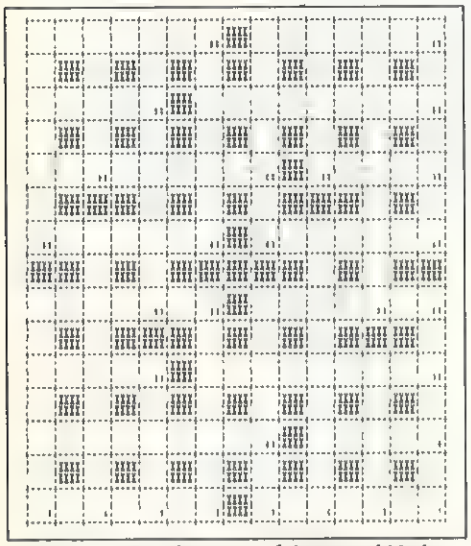


Figure 1 - Sample output of Crossword Modus

The main stages of crossword compilation are devising the grid, with special care for symmetry and various aspects of 'fairness'; forcing words into gaps, having as many tailored to the readership as possible; polishing this completed grid; entering the little numbers (harder than one might think); working up and polishing the clues; and producing and checking the fair copy.

Crossword Modus for PCs adds a useful degree of automation to most of these processes - at least to the ones that don't involve too much creativity. It draws the grid you want (rectangular, from 7x7 to 21x21); eases word entry; provides a pretty huge lexicon (over 1 MB) and uses it to find suitable words and phrases to fit the gaps; looks after symmetry; smooths the devising of clues (with, for instance, effective anagram searching); and prints out a fair copy (with those little numbers in place) as well as storing the final puzzle on disk. It's fairly easy to get into this process, and for sure using the program could make life a lot easier for crossword compilers. More importantly, it will make life much easier for people - such as teachers - who'd like to produce crosswords for their work but don't know how to go about getting professional results.

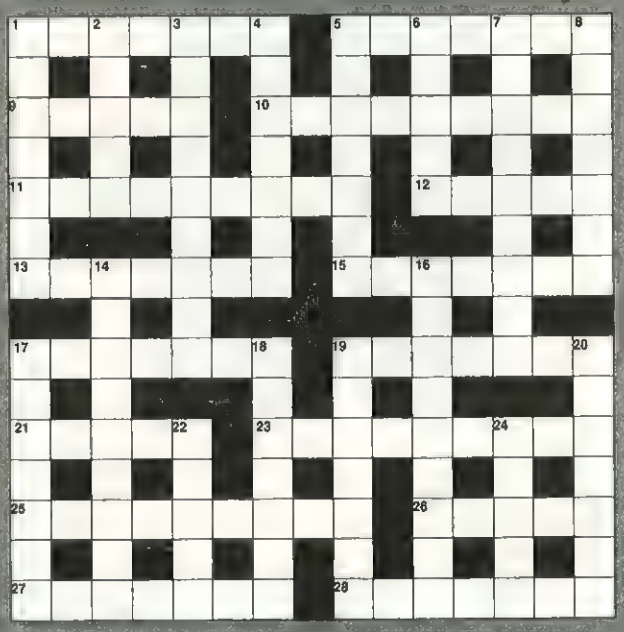
Crossword Modus also runs in two other modes. First, it allows the on-screen solving of puzzles saved on disk (again of clear value in education); the other mode uses the lexicon to sledge-hammer those 'how many words can you make from the letters of **.EXE MAGAZINE**?' exercises (109 of three or more letters, thanks for asking).

This £37.45 package is not going to put human crossword compilers out of business (I hope), but it certainly takes a lot of the drudgery out of what can be a very simulating pastime.

EXE

Eric Deeson has compiled .EXE's fiendishly tough crosswords since 1989. Crossword Modus is produced by Beamscan, 20 Vaughan Avenue, London NW4 4HU.

DECEMBER .EXEWORD



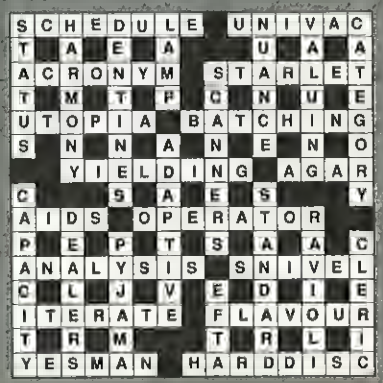
ACROSS

1 Like Christmas and New Year (7, 7)
5 See 1
9 Prepared message (5)
10 The science of old modems? (9)
11 They aim for perfection with a notion and data structures (9)
12 Beats of contract staff like 24? (5)
13 Puts on roll two way data structures (7)
15 Continues to iterate with spouse's boy (7)
17 Set down for common dbms? (7)
19 Concentrated the electron beam, for instance (7)
21 No initial meal from old cattle (5)
23 Prepare the data from non-standard form (9)
25 Keeps the system working (9)
26 Such rays can cause a crash (5)
27 Link in the jargon phrase (7)
28 Unhappy, sits round with cruel folk (7)

DOWN

1 Oxide of the core store (7)
2 Hardly autonomous network station (5)
3 I hear 11 with romantic rural views (9)
4 As time passes with oriental errors (7)
5 Non-working partners? (7)
6 Little donkey of use to the system perhaps (5)
7 Ancient oriental addictions? (9)
8 Shift on station? (7)
14 The one I/c the mainframe's data (9)
16 Swapped in a sort routine or Topic maybe (9)
17 Never constant ram, for instance (7)
18 Dealing with choppers, he stinted somehow (7)
19 Living communities to be saved by IT (7)
20 They ride compact discs (7)
22 Tease out bristles (5)
24 Ti-tum ti-tum ti-tum (5)

'.EXEWORD' compiled by Eric Deeson



NOVEMBER .EXEWORD

OUTSTANDING OPPORTUNITIES FOR '92

MAJOR UK CORPORATE

A leading UK corporation located alongside the M4 corridor has a number of exciting opportunities for 1992. This is a green-field opportunity for experienced people who have in-depth skills in the following areas:

C/Windows Developers

In-depth knowledge of 'C' and the Windows API is required along with the ability to communicate that knowledge to others. Network experience in the Windows environment would be a bonus, but is not essential.

LAN/WAN Specialists

We are looking for someone with detailed knowledge of the architectural issues and products available in this marketplace. The ability to design solutions which encompass DOS & Windows based PC's and IBM mainframes is fundamental. Knowledge of UNIX communications would be a bonus.

Candidates must be capable of demonstrating their involvement in the implementation of projects in the areas noted. Only people with in-depth knowledge and demonstrable skills need apply.

The appointments will be either on a permanent or contract basis. Remuneration is unlikely to be a problem for the right candidates. Permanent remuneration will be in the range £30k - £50k + car + benefits. Contract rates will be negotiable.

In the first instance, please call: 0793 641706

UNIX APPOINTMENTS

FT Associates

- UNIX
- DATABASES
- 4GLs
- OPEN SYSTEMS

Evening Telephone:
Gerard Fawcitt
081-341 7301
Marcus Langford-Thomas
081-681 3905
Facsimile: 071-487 4501

telephone
071-487 4110

S. Bucks £18,000 + Quality Car
Transputer company seeks experienced Technical Support person with relevant experience of Software and Hardware Support including some of the following; 'C', Helios, Express, Inmos tool sets.
Please quote reference: R. Fairclough

Hants & Lancs £25,000 + Car
Two Pre-Sales Support vacancies exist, applicants with good UNIX, Networking and Past Sales Support experience essential. Understanding of TCP/IP, Streams knowledge particularly useful. Client develops Document Management systems.
Please quote reference: A. Roberts

Berks £23,000 + Car
Software Engineer, minimum of 2 years Commercial experience, with Degree in Electronic Engineering and knowledge of C++/C and Structured Methodologies, particularly Yourdon. Must have a strong UNIX understanding, experience of VHDL or similar useful. Technical Support position also exists with a similar background.
Please quote reference: P. Sugden

Beds c£20,000 + Car + Benefits
System House, focussed on Central Government marketplace, seeks a Support Specialist. Essential skills are Oracle SQL, Forms; UNIX Support/problem solving, Pre/Post Sales, some Project Management.
Quote Reference: K. Barlow

S.W. London To £22,000
Systems integration Consultancy is keen to recruit 2 programmers. Projects include Software Development based around networking (TCP/IP, Token Ring), Porting Software to UNIX V.4, Applications Development. Essentials are excellent 'C' and UNIX (pref V.4) skills, and a flexible attitude. Windows 3/DOS also of interest.
Quote Reference: M. Platt

C. London £25,000
Software House and IBM VAR wishes to identify a Project Manager with a sound commercial background, with the ability to take projects through from specification to delivery. Essential skills are OS/2, Novell, LANS and WANS, and integration and build background.
Please quote Reference: A. Gilroy

LONDON - Analyst/Programmer Oracle, Pro*C, Form3, ... *£35k+*

LONDON - Soft/Engs - C, 68000, Z80 - Bios level..... *£19k+*

TOKYO - Team Leader - UNIX/R/T, Financial, 'C'..... *£22k + BB*

LONDON - A/P's Progress or other 4GL, Structured Methods..... *£23k*

LONDON - Soft/Eng - SUN, NFS, 'C' - Fin env ..... *To £25k*

LONDON - Sppt/Dev - SUN + Novell knowledge + 'C' *To £21k + BB*

BERKS - Soft/Dev - UNIX, T.P. experience + 'C' *£21k*

HERTS - Analyst/Programmer - Sybase, Uniface..... *£20k*

CAMBS - Soft/Eng - Low level UNIX, S/W & H/W exp. ..... *To £30k*

LONDON - Q/A - UNIX/Open System Project experience..... *£30k*

CAMBS - Technical Support - UNIX/Windows ..... *To £20k*

NOTTS - Tech. Support + Devt; Accts UNIX/Informix 4GL *c £25k + car*

E. Anglia - Oracle SQL Prog, & UNIX/C Programmer ..... *£ neg*

MIDDX - Proj Mgr, exp. large teams, C4 years UNIX, 2 yrs Oracle..... *c £27k*

CHESHIRE - Programmer - Degree C/MS-DOS, Windows, pref UNIX ..... *c £15k*

LONDON - Tech/Auth - UNIX/Open system experience..... *£25k*

FAWCITT THOMAS ASSOCIATES, 11 DUKE STREET, LONDON W1M 5RA

WEST YORKSHIRE

CLIPPER ANALYST PROGRAMMER to work in a dynamic & friendly environment undertaking new development work. Knowledge of 'C' an advantage.....*Salary to £16,000*

We currently have a very wide range of UNIX based vacancies for INGRES & COBOL under UNIX including project leaders, analyst programmers, pre/post sales support and systems admin.

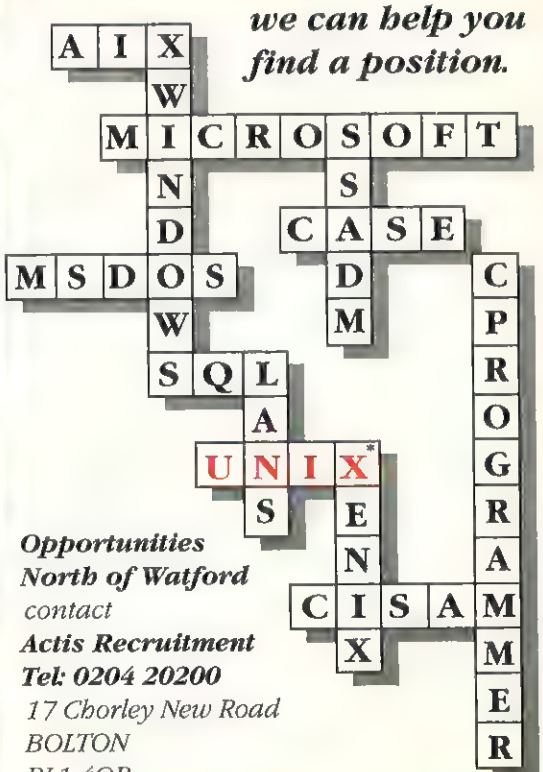
For your next career move around West Yorkshire, telephone **Vincent Atherton** on **Leeds (0532) 504560** or write to:

AIREDALE RECRUITMENT
Realtex House, Micklefield Lane, Rawdon, Leeds LS19 6AX

AIREDALE RECRUITMENT

ACTIS

If your skills are in the frame we can help you find a position.



Opportunities North of Watford
contact
Actis Recruitment
Tel: 0204 20200
17 Chorley New Road
BOLTON
BL1 4QR.

*Unix is a trademark of AT&T

ACTIS RECRUITMENT

'C' Analyst/Programmers
(learn Securities)

2 - 3 years experience
£18 - 24,000 + benefits

We are an international stockbroker with offices throughout the world, successful and still expanding. We currently have Unix and VAX systems and in London we are about to embark on a major new Front Office development using object-oriented techniques. Other major projects will follow.

We currently have a team of 4 on this project, but now seek a further 2 analyst/programmers to join the team. Probably graduates, you should have 2+ years 'C' or C++ and at least 6 months Unix. X-Windows/Motif, decision support, object-oriented techniques or experience in a financial environment would be a bonus.

You will need a flexible attitude, enjoy a technical challenge and hard work. If you meet this profile, we would like to hear from you.

We are in the City close to tubes and mainline stations. We have sports facilities and valuable benefits, **but more importantly we will give you the training and opportunity to further develop your skills.**

For more details or to apply call Bill Theobald on 071 489 0165 or evenings/ weekends on 0342 843278 or send him your CV.

Sector Personnel COMPUTER RECRUITMENT AND CONSULTANCY

12 Well Court London EC4M 9DN. Phone: 071-489 0165

Herts **to £25k**

Developing integrated corporate office software for the world-wide market place. You should have at least two of the following: **'C', UNIX, TCP/IP, X-Windows, GUI, E-Mail** or **Graphics.**

Bucks/Berks borders **to £23k**

Revolutionary developers of emulation technology seeking self-motivated software engineers for full project life-cycle development. If you have strong 'C' skills combined with some of: **UNIX, VMS, MacOS** or **MS-DOS.** 1st class employer offering profit share, share options and flexitime.

Yorks **£15k - £20k**

Forefront Spatial Information Developer is seeking innovative software designers with sound problem solving skills. Your skills should include 2 years of **'C'/UNIX** applications and experience of using relational database tools, any **structured methods, graphics** or **Windowing.**

City **to £30k + benefits**

Major financial institution is seeking bright software staff to develop **Real-Time Market Information Systems** covering such applications such as currency options, price modelling, risk analysis and time series databases. You should have solid 'C' skills combined with some of: **UNIX, VAX/VMS, Ethernet, C++, OS/2, Sybase, Windows, Motif** or **4GL's.** Benefits include subsidised mortgage!

MANY OTHER OPPORTUNITIES THROUGHOUT UK AND EUROPE

071 371-8496

ACUMEN SEARCH & SELECTION INTERNATIONAL
EBC House, 1a Ranelagh Gardens,
London SW6 3PA Fax: 071 371-8502
Out of hours telephone: 081 780-0637

ACUMEN

ASH ASSOCIATES

We specialise in the Recruitment of Software Design and Support Engineers in the South East for Real Time Applications including GRAPHICS COMMUNICATIONS, CONTROL/ROBOTICS, SIGNAL PROCESSING & MODELLING.

NEW YEAR - NEW START

The Start of the New Year sees several of our preferred clients needing to recruit bright young graduates with at least two years software experience particularly with 'C', Assemblers, MS DOS and UNIX gained within a High Technology environment.

Companies located in the SouthEast with specific needs for engineers with experience of the following applications including Medical Electronics, Television Control Systems, Telecommunications and Geographical Information Systems.

YOU will enhance your career and future prospects working within these High Technology Application areas with the backing and benefits of these small and expanding Companies.

Call James Hunt or Ron Cook NOW!
TEL: (0425) 475480 (24hrs) FAX: (0425) 480807

ASH associates

Recruitment Consultants
3 Pipers Ash, Ringwood, Hants, BH24 1UF

We would like to wish everyone who has made this year a success a MERRY XMAS & A HAPPY & PROSPEROUS NEW YEAR.

5-7 Sedley Place (off Oxford Street), London W1R 1HH Tel: 071 491 3640 Fax: 071 499 2546

IS2 - SCULPTOR - SQL

W.LONDON ***Up to £22K***

Any software engineer who has coded C for at least two years, and has used either query language, or IS2 or Sculptor in anger, would be very useful for this software house, they sell a top 5 product, and are looking for calibre individuals.

SCLBASE, ORACLE, DB2, INGRES

BUCKS ***£ GOOD***

A software engineer from a C coding background who is proficient in development of any of the above SQL driven products is required by a leading SQL products company - LAN & WAN or UNIX environments would also be advantageous.

OOPS & C++ SOFTWARE ENGINEER

W.LONDON ***£ GOOD***

Experience of either JPI, Zortech or Borland product versions would be immaterial, as long as you understand the C++ methodology and can code it with ability. A good development in a good systems house.

C - OS/2 - SOFTWARE ENGINEER

LONDON ***Up to £25K***

Good C development in an OS/2 environment. A quality software house involved in environment level systems, porting their product from DOS to OS/2, PM or AM experience is a distinct bonus.

C - UNIX - X-WINDOWS

CAMBS ***£ GOOD***

A software house with a reputation of good X-Windows development is seeking two good software engineers. They must both have at least 2-3 years experience, and a flexible ability - good structured C is an absolute must.

REALTIME C UNDER DOS

LONDON ***£ GOOD***

A good project leader who is young and enthusiastic, or a senior software engineer at the least, is required to run a small team in this software house. It is a comms application and has a GUI front end - so this kind of experience would be useful.

WINDOWS & C SOFTWARE ENGINEERS

HOME COUNTIES ***Up to £30K***

You must have good development skills in a windows SDK environment, either PC or UNIX platforms are equally welcome. There are some interesting GUI applications and some heavy duty C coding available, good prospects are assured in this renowned systems house.

MAINFRAME/PC PRODUCTS CONSULTANT

BERKS ***£ High plus substantial benefits***

Good systems skills & connectivity experience are essential prerequisites for this job. You should have good mainframe tool sets, preferably in an MVS environment; CICS; TELON; DB2; ACF; TSO; JCL; COBOL etc with either PC or UNIX systems development as well. A good opportunity for either mainframe system programmers to 'downsize' or PC/UNIX software engineers to broaden their aspect.

C - APPLE MAC - PC SYSTEM PROGRAMMER

LONDON ***Up to £20K***

New company that is developing a windowing product. Good systems level developer required. MPW C; Postscript; TIF; PIF etc. An interesting project for the right individual.

This is only a small selection of what is currently available. Please do not hesitate to give either Mike Dearing or Simon Gudgeon a call on 071 491 3640, or after hours on 081 767 1003

WANTED

to

Develop high performance networking products:

- **Software Engineers**
- **Hardware Engineers**

Madge Networks, the world leader in token ring networks for PCs, needs more excellent engineers. Our engineers:

- Develop World Class Products
- Have fun in the process
- Form a highly-motivated team
- Make a major contribution to a company doubling in size each year.

Personal excellence is an absolute requirement. All applicants will have at least one of:

- A First Class Honours Degree
- 3 A grades in numerate A-level subjects
- A world class flair for developing computer systems

Previous experience of networking is not required. Starting salaries for engineers range up to £30K, depending on experience.

To apply, please send a full CV to Fiona Woods at:

MADGE

Madge Networks Ltd
100 Lodge Lane
Chalfont St. Giles
Bucks HP8 4AH

ADVERTISERS INDEX

STOB - About...

Understandably, many of you have been writing to Ms Stob, known in her hairdresser's as 'the Petzold of Hounslow', to ask her to solve your Windows problems.

Dear Verity,
The enclosed photograph shows an icon which I found on my desktop. It does not seem to be attached to any particular application, and I cannot make out what it is supposed to represent. When you click on it, the hard disk runs for a couple of seconds, but there are no other observable effects. Please can you help?
RN, Epping.

Unfortunately RN's photograph was not up to the high standard that .EXE's production department requires, so I will draw you a word picture. Imagine a two-dimensional half-brick, done in green, on the back of a doughnut, rotated through 30 degrees counter-clockwise, on a grey background. This is an escapee from one of those floating blocks of incomprehensible button icons that are now de rigueur in Windows apps. It is clearly a visual metaphor for 'transposé the current spreadsheet relative to the current marked block' or, if you haven't got a spreadsheet, it could be 'reload default monochrome palette from network drive without discarding current drawing'. Anyway, RN, it's not doing you any harm, so stop being a bully and leave it alone.

Dear Verity,
Over recent months I have read many articles describing OLE, and have formed the distinct impression that nobody knows what it is, or what it's for; although many journalists expend a lot of effort trying to pretend that they do understand. May I rely on you to do better?
HU, Witham.

Yes. Next please.

Dear Verity,
For many years we have been making a good living from a very poorly designed program which does little and is difficult to use. We also make a lot of money by selling very expensive support contracts. Now we are very worried. GUIs are going to make the function of programs clear, and their usage straightforward. Does the increasing popularity of Windows spell the end of the road for us?
JL, Southend.

Stop panicking, JL. It is absolutely no trouble to write difficult-to-use programs under Windows. Here are a few ideas to set you on the right road. 1) All your drop down menus should contain at least nine items. Six of these should always be greyed out. 2) Provide full Windows Help, but include only a page-and-one-half of not-very-helpful text. (You might think that this is being too positive, but remember that Windows Help takes five seconds to load, and messes up the desktop). If you are feeling especially daring, you may even like to open the help file of a completely different program, eg the rotten Solitaire card game. 3) If you are aiming to get up the nose of the punter, you could do far worse than write an MDI app. Child windows can easily be placed extremely awkwardly within the parent, and the profusion of maximise/minimise corner icons soon baffles the punter. See the Windows 3.0 File Manager for an eg. 4) Add a free-floating icon bar thingy ® as described in my answer to RN.

Dear Verity,
Have you noticed how many Windows-type acronyms consist of three letters; eg GUI, OLE, UAE? Don't you find this rather exciting and more than a little sinister?
TE, Basildon.

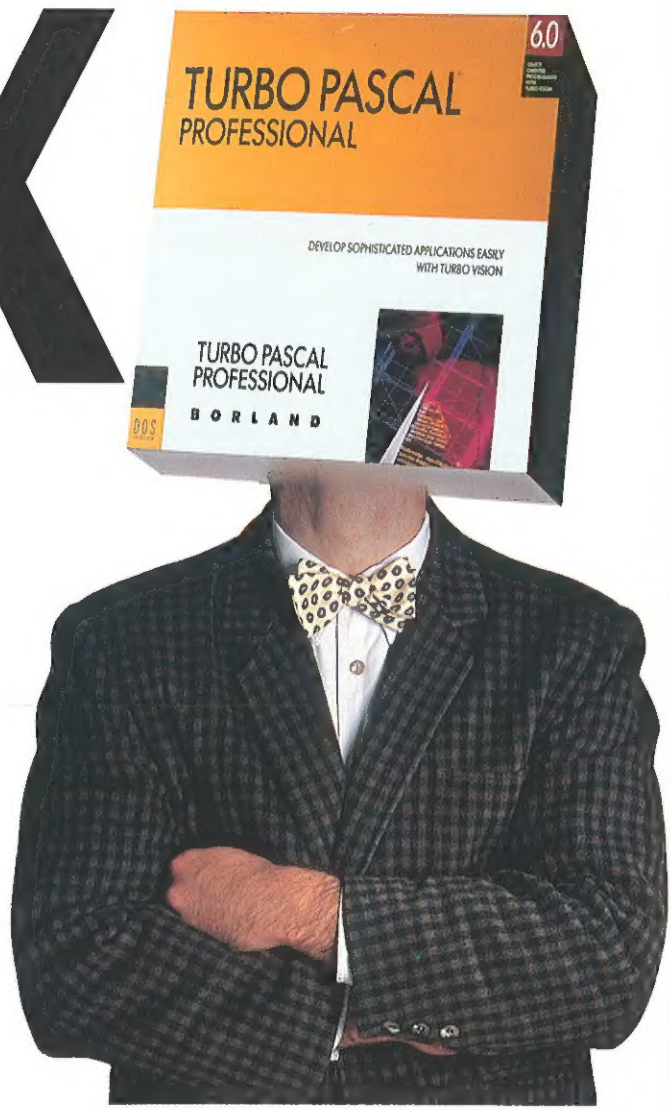
No.

Dear Verity,
Sorry, I have to pick you up on this. In the answer before last, you cast aspersions on the excellent Windows Solitaire game. My wife and I have enjoyed many very pleasant evenings playing this clever game. Why were you rude about it?
PP, Chelmsford.

Because, unlike real Solitaire, you can't cheat.

.EXE

BRAIN BOX



From inspiration to application – the brightest way

BORLAND'S new **Turbo Pascal® 6.0** with **Turbo Vision™** is the World's Number One Pascal compiler. Now you can concentrate on functionality and performance, rather than spending time on program mechanics and interfaces.

Turbo Vision is a revolutionary object-oriented application framework for MS™-DOS. There's no need to keep writing and debugging interface code – eliminating the risk of introducing new bugs to a proven sub-system. Turbo Vision brings a rational approach to program design with its single architectural structure.

Even better, **Turbo Pascal 6.0 Professional** gives you not only a Turbo-driven compiler that fully uses your extended memory, but also the World's fastest 100% MASM-compatible assembler, the World's first interactive profiler, and **Turbo Debugger®2.0**, which supports OOP.

If all that doesn't make you think, think about this too! Turbo Pascal 6.0 Professional retails for a remarkably competitive £199.95 plus VAT, and Turbo Pascal 6.0 for £99.95 plus VAT. And that's not all – if you're an existing Turbo Pascal user, take advantage **now** of Borland's very special upgrade offers.

For more information, contact your Borland dealer, ring Borland Customer Services on 0734 320022, or send the Freepost coupon today.

You'll never make a smarter move!

BORLAND
SOFTWARE
KNOW HOW

Borland International (UK) Ltd
8 Pavilions, Ruscombe Business Park
Twyford, Berkshire RG10 9NN Telephone 0734 320022

CIRCLE NO. 410

From the makers of Paradox®, Quattro® Pro, Sidekick®, Turbo Pascal® and Turbo C++®

NAME _____

JOB TITLE _____

COMPANY _____

ADDRESS _____

_____

POSTCODE _____ TEL _____

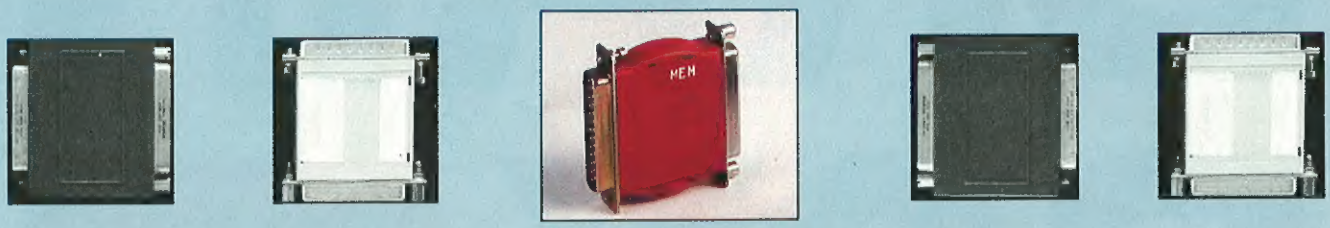
I currently use Turbo Pascal ☐

.EXE 12/91

Please return this coupon to: Turbo Pascal 6.0 Information, Borland International (UK) Ltd, FREEPOST RG1571, Twyford, Berkshire RG10 8BR

SOFTWARE PROTECTION!

Can You Spot The Difference?



- ☑ High-level security keys
- ☑ Assembler-based, customer specific, encrypted interrogation routines
- ☑ Over 140 language interfaces available
- ☑ Compatibility - due to 9 years experience & 750,000+ keys sold
- ☑ Reliable on-going support
- ☑ Physically unique keys for each customer
- ☑ MS-DOS, OS/2, UNIX, XENIX, WINDOWS 2 & 3, "MACINTOSH"
- ☑ Transparent operation with most peripherals on market
- ☑ Free language updates
- ☑ Parallel, Serial, Mac ports

OVER 3,500 CUSTOMERS WORLDWIDE HAVE!

Electronic Key

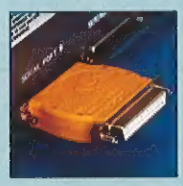
The ideal device for identically produced software packages. Uniquely wired with customer code and a software code. Uses Assembler based program, decryption interface and random values.

1 Word Memory Key

Custom hardware wiring allows the developer total control over information stored in the key. 2 bytes of memory allows several packages to be protected with just the one key.

31 Word Memory Key

For multiple software protection schemes. 31 words of 16 bits of non-volatile dynamically programmable memory. Its capacity to store information provides virtually limitless power. Flexible protection scheme can be modified on-site during operation of software package.



Micro Processor Key

Provides the ultimate in software security. Not tied to any language or O/S. 8 bit microprocessor powers from RS-232 level. Requires no power supply. For PC terminals, minis, & others using RS 232 C comms. Used on workstations. This key is effectively a computer.

Macintosh Key

Extremely powerful & customised protection for the Mac. 31 words available for random storage. Providing unequalled protection the Macintosh Memory Key connects to the SCSI 25 pin port and operates transparently.

MICROPHAR... The European Leader*

* Based on the number of keys in use throughout Europe

For a no-nonsense informative discussion on how our dongles can better protect your profits, please call us on:

091-378 9191

CLEARSOFT

CLEARSOFT, Littleburn Ind. Estate,
Langley Moor, Durham DH7 8HG.
Telephone: (091) 378 9191 Fax: (091) 3789393

Please send me full details on your protection systems.

Name: .....

Position: .....

Address: .....

.....

Telephone: .....

Fax: .....

Signature: .....

Clearsoft, Littleburn Ind Est,
Langley Moor, Durham, DH7 8HG
Tel: 091 3789191 Fax: 091 3789393

.EXE 12/91

CIRCLE NO. 411